



Universidade Federal
do Rio de Janeiro
Escola Politécnica

Projeto e Construção de Protótipo de Sistema de Comunicação por Luz Visível (VLC)

Arthur Antunes Braga Bedor

Projeto de Graduação apresentado ao Curso de Engenharia Eletrônica e de Computação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: M.Sc. Mauros Campello Queiroz

Rio de Janeiro

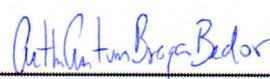
Fevereiro de 2017

Projeto e Construção de Protótipo de Sistema de Comunicação por Luz Visível (VLC)

Arthur Antunes Braga Bedor

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRÔNICO E DE COMPUTAÇÃO

Autor:



Arthur Antunes Braga Bedor

Orientador:

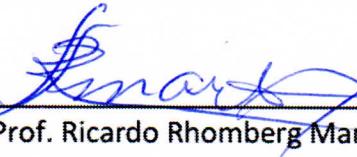

Prof. Mauros Campello Queiroz, M.Sc.

Examinador:



Prof.ª Mariane Rembold Petraglia, Ph.D.

Examinador:


Prof. Ricardo Rhomberg Martins, D.Sc.

Rio de Janeiro – RJ, Brasil

Fevereiro de 2017

Declaração de Autoria e de Direitos

Eu, Arthur Antunes Braga Bedor CPF: 147.656.047-13, autor da monografia “Projeto e Construção de Protótipo de Sistema de Comunicação por Luz Visível (VLC)”, subscrevo para os devidos fins, as seguintes informações:

1. O autor declara que o trabalho apresentado na disciplina de Projeto de Graduação da Escola Politécnica da UFRJ é de sua autoria, sendo original em forma e conteúdo.
2. Excetuam-se do item 1. eventuais transcrições de texto, figuras, tabelas, conceitos e ideias, que identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos proprietários, quando necessárias.
3. O autor permite que a UFRJ, por um prazo indeterminado, efetue em qualquer mídia de divulgação, a publicação do trabalho acadêmico em sua totalidade, ou em parte. Essa autorização não envolve ônus de qualquer natureza à UFRJ, ou aos seus representantes.
4. O autor pode, excepcionalmente, encaminhar à Comissão de Projeto de Graduação, a não divulgação do material, por um prazo máximo de 01 (um) ano, improrrogável, a contar da data de defesa, desde que o pedido seja justificado, e solicitado antecipadamente, por escrito, à Congregação da Escola Politécnica.
5. O autor declara, ainda, ter a capacidade jurídica para a prática do presente ato, assim como ter conhecimento do teor da presente Declaração, estando ciente das sanções e punições legais, no que tange a cópia parcial, ou total, de obra intelectual, o que se configura como violação do direito autoral previsto no Código Penal Brasileiro no art.184 e art.299, bem como na Lei 9.610.
6. O autor é o único responsável pelo conteúdo apresentado nos trabalhos acadêmicos publicados, não cabendo à UFRJ, aos seus representantes, ou ao(s) orientador(es), qualquer responsabilização/ indenização nesse sentido.
7. Por ser verdade, firmo a presente declaração.



Arthur Antunes Braga Bedor

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Escola Politécnica - Departamento de Eletrônica e de Computação
Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária
Rio de Janeiro - RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es).

Bedor, Arthur Antunes Braga

Projeto e Construção de Protótipo de Sistema de Comunicação por Luz Visível (VLC) / Arthur Antunes Braga Bedor. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2017.

XVI, 60 p.: il.; 29,7 cm.

Orientador: Mauros Campello Queiroz

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia Eletrônica e de Computação, 2017.

Referências Bibliográficas: p. 46-47.

1. Comunicação Sem Fio
 2. Comunicação Óptica
 3. Comunicação por Luz Visível (VLC).
 4. Light Fidelity (Li-Fi).
 5. OFDM
- I. Queiroz, Mauros Campello. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Eletrônica e de Computação. III. Título.

A ignorância é a maldição de Deus. O conhecimento, a asa com a qual nos elevamos aos céus.

William Shakespeare, Rei Henrique VI

Dedico este trabalho aos meus familiares, que sempre me apoiaram, e à comunidade acadêmica e científica, caso este projeto possa colaborar de alguma forma em sua nobre e incansável busca por conhecimento.

AGRADECIMENTOS

*Há coisas que não se pode fazer junto
sem acabar gostando um do outro, e
derrubar um trasgo montanhês de quase
quatro metros de altura é uma dessas
coisas.*

J. K. Rowling,
Harry Potter e a Pedra Filosofal

Existe uma teoria filosófica que defende que todo acontecimento pode ser explicado através de relações de causalidade. De forma geral, o determinismo, como é conhecida a teoria, se baseia no princípio de que toda ação gera uma reação, resultando em uma complexa e infinita rede de cadeias de causa e efeito que, à princípio, é impossível de ser identificada pelo ser humano. Um cometa cruzando o sistema solar tem seu curso alterado pela enorme atração gravitacional de Júpiter, e entra em rota de colisão com a Terra, extinguindo os dinossauros e permitindo o desenvolvimento de uma nova espécie de mamífero que, após diversas gerações naturalmente selecionadas, dão origem ao ser humano. Tudo parte deste complexo sistema causal chamado Universo, cujas condições iniciais representam o Santo Graal da humanidade, a busca por respostas.

Portanto, seguindo a lógica determinista, por maior que seja a tentação de dizer que este presente trabalho é meu, não seria capaz de fazê-lo. Este projeto não é apenas meu, mas também de todos aqueles que colaboraram para que hoje, depois de anos de estudo e comprometimento, eu estivesse escrevendo esses agradecimentos. Pois todos os projetos finalizados, todas as conquistas alcançadas – e alguns fracassos também que, afinal, fazem parte do aprendizado – foram frutos colhidos da dedicação, do entusiasmo, de escolhas e da ajuda de terceiros, nessa entrelaçada relação causal que tem sua conclusão neste trabalho. E seria de imensa ingratidão minha não valorizar o apoio, os conselhos, as dicas e a amizade daqueles que fizeram parte de todo esse processo. Por isso, agradeço:

A Universidade Federal do Rio de Janeiro, por proporcionar um ambiente de atividades diversificadas, que agregam aos alunos experiências que vão além da formação profissional.

Ao corpo docente do DEL, cuja excelência muitas vezes se revelou inspiradora. Em especial ao Professor Casé, por sua inabalável dedicação aos alunos do curso de Eletrônica.

Aos professores da Universidade que se empenharam em ensinar que ser engenheiro não significa abdicar da História, da Arte e da responsabilidade social.

Ao meu orientador Mauros Campello, por aceitar o compromisso tão em cima da hora e pelo suporte no pouco tempo que lhe coube. Ao meu primeiro orientador Gelson Mendonça, por ter aceitado me orientar mesmo com seus problemas de saúde. E ao professor Wallace Martins pela ajuda com as dúvidas no projeto.

Aos professores do CEFET, que me iniciaram nesta jornada rumo à Engenharia, e também na formação de um senso crítico mais apurado.

Ao meu amigo e parceiro de aulas, Geovane, e aos amigos da Portugal Telecom Inovação, Gabriel e Marcelo, pela amizade, conselhos e dicas que me ajudaram a suportar e vencer os desafios que surgiram, além das discussões saudáveis que sempre mantivemos, e que contribuíram muito para o meu crescimento pessoal.

E, por fim, aos meus familiares, por todo o apoio e carinho que têm dedicado a mim em todos esses anos. A minha irmã Nathalia, por me motivar à leitura. Ao meu pai, por me ensinar o valor de uma boa educação e da disciplina nos estudos. A minha mãe, por me apoiar e torcer por minhas escolhas. E a minha irmã Priscilla, por ter sido minha co-orientadora informal e ter me auxiliado bastante na confecção deste documento.

Este trabalho foi feito por vocês.

RESUMO

O espectro de radiofrequência tem se tornado cada vez mais escasso nos últimos anos devido ao aumento exponencial do número de dispositivos móveis que utilizam as redes sem fio para se conectar com a internet. Por isso, novas formas de estabelecer conexão sem fio, utilizando outras faixas do espectro de frequência, vêm sendo desenvolvidas. Uma delas se baseia nas comunicações por luz visível, ou VLC (*Visible Light Communications*): tecnologia na qual a variação da intensidade da luz é responsável pela transmissão dos dados.

A proposta deste trabalho é elaborar, como prova de conceito, um protótipo de um sistema VLC para envio de dados em uma arquitetura unidirecional ponto-a-ponto, utilizando-se um sistema OFDM para a recuperação de informações perdidas devido às características do canal de comunicações. Também será apresentado um breve estado-da-arte da tecnologia de comunicação por luz visível, suas vantagens em relação às outras tecnologias sem fio e suas possíveis aplicações nas futuras redes 5G, como o Li-Fi (*Light Fidelity*).

Palavras-chave: Comunicação Sem Fio, Comunicação Óptica, Comunicação por Luz Visível (VLC), Light Fidelity (Li-Fi), OFDM

ABSTRACT

The radiofrequency spectrum has become increasingly scarce in the last years due to the exponential growth of the number of mobile devices that use wireless networks to connect themselves to the internet. For this reason, new ways of establishing wireless connection in other frequency bands of the spectrum has been developed. One of them is based in the visible light communication, or VLC: a technology in which the variation of light intensity is responsible for the data transmission.

The purpose of this work is to develop, as proof of concept, a prototype of a VLC system which sends data in unidirectional peer-to-peer architecture, using an OFDM system for recovery of missing information caused by communication channel characteristics. Also, it will be presented a brief state-of-art of visible light communication technology, its advantages in comparison to other wireless technologies, and its possible applications in the upcoming 5G networks, as Li-Fi (Light Fidelity).

Keywords: Wireless Communication, Optical Communication, Visible Light Communication (VLC), Light Fidelity (Li-Fi), OFDM

SIGLAS

3G	3ª Geração
4G	4ª Geração
5G	5ª Geração
ADC	Analog-to-Digital Converter
BER	Bit Error Rate
BPL	Broadband Over Power Line
BPSK	Binary Phase Shift Keying
DAC	Digital-to-Analog Converter
DCO-OFDM	Direct-Current Biased Optical OFDM
DD	Direct Detection
DFT	Discrete Fourier Transform
ERB	Estação Rádio Base
FDD	Frequency-Division Duplexing
FDM	Frequency-Division Multiplexing
IBI	InterBlock Interference
IEEE	Institute of Electrical and Electronics Engineers
IM	Intensity Modulation
IR	Infrared Radiation
ISI	InterSymbol Interference
ISM	Industrial, Scientific and Medical
ITU	International Telecommunication Union
IDFT	Inverse Discrete Fourier Transform
LED	Light Emitting Diode
Li-Fi	Light Fidelity
MAC	Media Access Control
MIMO	Multiple-Input Multiple-Output
MMSE	Minimum Mean Square Error
OFDM	Orthogonal Frequency-Division Multiplexing
OOK	On-Off Keying
OWC	Optical Wireless Communication

PAM	Pulse-Amplitude Modulation
PPM	Pulse-Position Modulation
PWM	Pulse-Width Modulation
QAM	Quadrature Amplitude Modulation
RF	Radiofrequênciā
SISO	Single-Input Single-Output
SNR	Signal-to-Noise Ratio
TDD	Time-Division Duplexing
VLC	Visible Light Communication
VoD	Video On Demand
WDD	Wavelength-Division Duplexing
Wi-Fi	Wireless Fidelity

Sumário

1.	Introdução.....	2
1.1	Tema.....	2
1.2	Justificativa	2
1.3	Objetivos	3
1.4	Resumo dos Capítulos	4
2.	Comunicação por Luz Visível	6
2.1	Histórico	6
2.2	VLC e Li-Fi	9
2.3	Li-Fi vs. Wi-Fi.....	12
3.	Projeto.....	18
3.1	OOK (<i>On-Off Keying</i>)	19
3.2	DCO-OFDM (<i>DC-Biased Optical OFDM</i>).....	20
3.3	Transmissor Óptico	27
3.3.1	DAC (Conversor Analógico-Digital).....	27
3.3.2	Driver LED	29
3.3.3	Algoritmo de Transmissão.....	31
3.4	Receptor Óptico	32
3.4.1	Sensor de Luminosidade	32
3.4.2	Algoritmo de Recepção	33
4.	Resultados e Discussão	36
4.1	Simulação	36
4.2	Experimento Prático.....	38
4.2.1	Distância entre módulos	39
4.2.2	Período de Símbolo (Taxa de Transmissão)	40
4.2.3	Envio de Imagem	41
5.	Conclusão	44
	Referências.....	46
	ANEXO A.....	48
	ANEXO B	52
	ANEXO C	54
	ANEXO D	57
	ANEXO E	59

Lista de Figuras

Figura 1 – Previsão do crescimento mundial do número de dispositivos móveis.....	2
Figura 2 – Telégrafo Óptico de Claude Chappe.....	7
Figura 3 – Módulo transmissor (esquerda) e receptor (direita) do fotofone de Graham Bell.	8
Figura 4 – Espectro de radiação eletromagnética.....	9
Figura 5 – Sistema VLC genérico, com blocos transmissor e receptor apresentados.....	10
Figura 6 – Cenário de uso da tecnologia Li-Fi.....	11
Figura 7 - Principais camadas do Li-Fi e VLC e áreas de aplicações.	12
Figura 8 - Comparação entre acesso à internet via Wi-Fi e via Li-Fi.	13
Figura 9 - Esquema simplificado de montagem do protótipo de sistema VLC.	18
Figura 10 - Transmissão Óptica OOK.....	19
Figura 11 - Diagrama de blocos de transmissão óptica usando OOK.....	20
Figura 12 - Representação das subportadoras OFDM no domínio da frequência.....	21
Figura 13 - Inserção do Prefixo Cílico no período de guarda entre símbolos OFDM.	22
Figura 14 - Diagrama de Blocos de um sistema linear invariante no tempo.	22
Figura 15 - Diagrama de Blocos OFDM simplificado.	23
Figura 16 - Transmissor OFDM usando simetria hermitiana para comunicação óptica	25
Figura 17 - Diagrama de blocos completo do sistema OFDM.....	26
Figura 18 - Transmissão DCO-OFDM: (a) Sem DC-Bias (b) Com DC-Bias.....	27
Figura 19 - Modulação PWM no ATmega328 com variação de duty cycle.....	28
Figura 20 - DAC utilizando PWM e filtro RC passa-baixas.....	29
Figura 21 – Características do LED: (a) Distribuição Espectral (b) Diagrama de Radiação.....	29
Figura 22 – Circuito completo do Transmissor Óptico com Driver.	30
Figura 23 – Algoritmo do código de transmissão.....	31
Figura 24 - Diagrama do circuito interno do CI OPT101.	32
Figura 25 - OPT101: (a) Resposta Espectral (b) Resposta ao Ângulo de Incidência.	33
Figura 26 - Algoritmo do código de recepção.	34
Figura 27 - Diagramas de constelação do sistema OFDM simulado: (a) BPSK (b) 4-QAM.....	36
Figura 28 - Gráfico de BER x SNR para sistema OFDM.	37
Figura 29 - Foto dos módulos transmissor (esquerda) e receptor (direita).	38
Figura 30 - Gráfico BER x Distância para os sistemas OOK e OFDM.	39
Figura 31 - Gráfico BER x Período de símbolo para os sistemas OOK e OFDM.	40
Figura 32 - Imagens enviadas pelos sistemas OOK (a e b) e OFDM (c e d). Para (a) e (c) o período de símbolo utilizado foi 500 µs, e para (b) e (d), 600 µs.....	42

Lista de Tabelas

Tabela 1 - Taxas de dados alcançáveis pelas tecnologias Wi-Fi em comparação com o Li-Fi.....	14
Tabela 2 - Tabela comparativa dos resultados obtidos com os sistemas OOK e OFDM.....	42

Capítulo 1

Introdução

1. Introdução

1.1 Tema

Este trabalho consiste na elaboração e montagem de um protótipo de sistema de comunicação sem fio, utilizando o conceito da comunicação por luz visível, também conhecida como VLC (*Visible Light Communication*). Tal tecnologia se apresenta como uma das possíveis soluções para as comunicações sem fio, cujo espectro de radiofrequência vem se tornando cada vez mais saturado.

1.2 Justificativa

O número de dispositivos conectados à internet tem crescido exponencialmente nas últimas décadas (COMPTIA, 2015), muito devido à popularização dos dispositivos móveis, como smartphones, tablets e notebooks, que utilizam principalmente as redes sem fio para se comunicar com a internet. Atrelada a isso está a demanda crescente por largura de banda, visando prover serviços que necessitam de uma conexão de alta velocidade, como serviços de streaming e VoD (*Video on Demand*). Projeções indicam que até 2021 cerca de 11,6 bilhões de dispositivos móveis estarão conectados à internet (Figura 1) e de que o volume de dados transferidos mensalmente tenderá a ser cada vez maior, levando à eventual escassez do espectro de radiofrequência (CISCO, 2017).

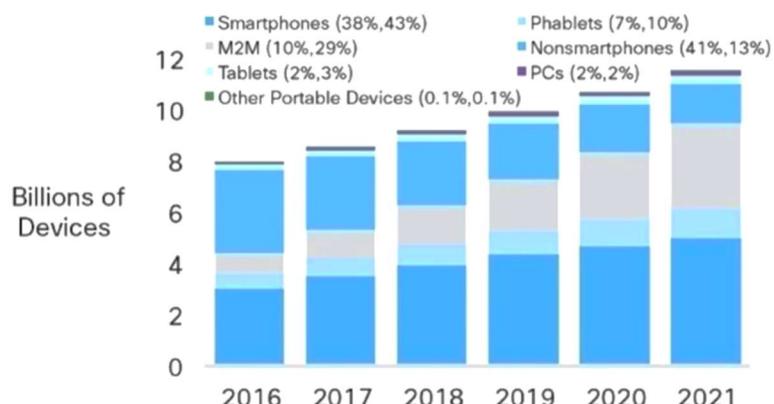


Figura 1 – Previsão do crescimento mundial do número de dispositivos móveis.
Fonte: CISCO, 2017.

Por isso, novas formas de se estabelecer conexão sem fio, utilizando outras faixas do espectro de frequência, vêm sendo desenvolvidas. Uma delas se baseia nas comunicações por luz visível, ou VLC (*Visible Light Communications*), tecnologia na qual a variação da intensidade da luz é responsável pela transmissão dos dados. A grande vantagem desse tipo de comunicação é a possibilidade de alcançar altas taxas de transmissão, já que possui uma maior largura de banda do que sistemas de rede sem fio Wi-Fi e outras comunicações via rádio convencionais. Além disso, trabalha numa faixa do espectro ainda pouco explorada para comunicações, o que evita interferências com outros sinais. Também permite que a mesma luz que transmite informação possa ser usada como luz ambiente, já que a alta frequência na qual a informação é transmitida não é perceptível à visão humana. Como ondas eletromagnéticas nessa faixa de frequência não podem atravessar paredes ou objetos sólidos, o sistema é também uma forma de comunicação mais segura à interceptação de sinal, o que pode ser útil em determinadas aplicações como, por exemplo, no meio militar.

Essa forma de comunicação, que ainda está em fase de desenvolvimento, tem avançado bastante nos últimos anos e faz parte do conjunto de tecnologias consideradas promissoras para a realização das redes móveis de 5ª Geração, ou 5G (WANG *et al.*, 2014). Além disso, a tecnologia em desenvolvimento conhecida como Li-Fi (*Light Fidelity*), que se baseia nas comunicações por luz visível, vem trazendo novas especificações com o objetivo de se alcançar um padrão, tal qual é o do hoje popular Wi-Fi. Portanto, a importância do trabalho proposto reside na experimentação da técnica que potencialmente terá grande presença no futuro das telecomunicações.

1.3 Objetivos

O objetivo deste trabalho é elaborar um protótipo de um sistema de comunicação no qual o conceito da comunicação por luz visível possa ser validado. Os objetivos específicos incluem: (1) enviar dados entre dois pontos através do ar utilizando a variação da intensidade da luz. (2) utilizar métodos de codificação de linha e de canal para a transmissão dos dados. (3) realizar o processamento do sinal recebido, recuperando dados que possam ter se perdido na transmissão, e a decodificação dos dados recebidos para poderem ser interpretados pelo usuário.

1.4 Resumo dos Capítulos

No Capítulo 2 será apresentado um breve histórico sobre as comunicações ópticas e abordados o estado-da-arte das comunicações por luz visível, suas vantagens em relação às outras tecnologias sem fio e sua importância no futuro cenário das telecomunicações. No Capítulo 3 será apresentada a metodologia para a formulação do projeto, incluindo os materiais e técnicas utilizados para sua execução. Os resultados dos experimentos serão apresentados e discutidos no Capítulo 4. O Capítulo 5 contará com a conclusão e propostas de melhorias para o projeto.

Capítulo 2

Comunicação por Luz Visível

2. Comunicação por Luz Visível

A comunicação por luz visível (VLC) pode ser considerada um caso especial das comunicações ópticas sem fio (*Optical Wireless Communications* - OWC), já que essa utiliza uma pequena região do espectro de frequência, correspondente ao espectro visível, para carregar informação. Como a transmissão é feita pela luz de forma não guiada, ou seja, sem um meio que conduza a luz do ponto de origem ao ponto de destino, as OWCs podem ainda ser consideradas casos especiais das comunicações ópticas em geral, dentre as quais se enquadraria, por exemplo, a transmissão de dados via fibra óptica.

Para abordar as características de um sistema VLC e suas aplicações é preciso compreender a diferença entre as variadas formas de comunicação óptica, assim como as principais diferenças entre essas e os sistemas sem fio usuais, como o rádio e o Wi-Fi. Neste capítulo, será apresentado um breve histórico do desenvolvimento das comunicações ópticas e as principais características dos sistemas VLC, bem como descrever sobre a tecnologia em desenvolvimento conhecida como Li-Fi, as vantagens e desvantagens do Li-Fi em relação ao Wi-Fi, e as possíveis aplicações dessa tecnologia na vida cotidiana.

2.1 Histórico

O uso da luz como forma de comunicação a longa distância remonta milênios. Antigas civilizações, como a Grega e a Romana, já empregavam sinais de fumaça ou grandes fogueiras para enviar mensagens sobre cercos e batalhas ou alertar às defesas de que uma tropa inimiga se aproximava. A Ilíada, poema épico escrito por Homero cerca de 700 a.C e que descreve os acontecimentos da Guerra de Tróia, contém as primeiras referências sobre o uso de faróis de fogo ou de fumaça para fins militares. Em Agamenon, outro poema épico escrito por Ésquilo em 458 a.C., é descrito como a mensagem sobre a queda de Tróia foi enviada por Clitenestra, esposa de Agamenon, para a cidade de Argos através de uma cadeia de faróis de fogo, percorrendo uma distância total de aproximadamente 480 km (BURNS, 2004).

Outros métodos mais engenhosos, envolvendo formas de codificação utilizando tochas, foram propostos pelo historiador grego Políbio e pelo historiador cristão Sexto Júlio Africano séculos depois. Porém, foi apenas em 1792 que um sistema prático de telégrafo óptico foi estabelecido em escala nacional pelo inventor francês Claude Chappe. Inspirado pelos trabalhos dos ingleses Robert Hooke e Edward Somerset, Chappe implantou uma rede de 556 estações de semáforos espalhadas pela França, cobrindo uma distância total de 4800 km. Mensagens podiam ser enviadas da cidade francesa de Toulon e serem recebidas em Paris em 13 minutos e 50 segundos (BURNS, 2004).

O semáforo de Chappe (Figura 2) consistia de uma coluna de madeira, chamada regulador, de aproximadamente 4,5 metros de comprimento e 0,35 metros de largura, onde eram fixados dois braços, chamados indicadores, de aproximadamente 2 metros de comprimento. Os indicadores e o regulador eram erguidos em torres e podiam ser posicionados de maneiras diferentes, podendo representar até 196 símbolos. Com o uso de telescópios, a estrutura podia ser observada a quilômetros de distância e a mensagem era enviada de semáforo em semáforo até o destino. Esse sistema funcionou em diversos países por muito tempo, até que foi substituído pelo telégrafo elétrico (HUUDEMAN, 2003).

O heliógrafo - inventado por Sir Henry C. Mance em 1869 - foi outro instrumento que se utilizou da codificação da luz, adaptando o helióstato de forma a enviar mensagens codificadas em Morse a grandes distâncias pelo chaveamento manual da luz do sol refletida por seu espelho. Suas maiores vantagens eram a portabilidade, baixo custo, longo alcance e segurança quanto ao segredo da mensagem, já que apenas observadores em visada direta com o transmissor poderiam captar a mensagem (BURNS, 2004).

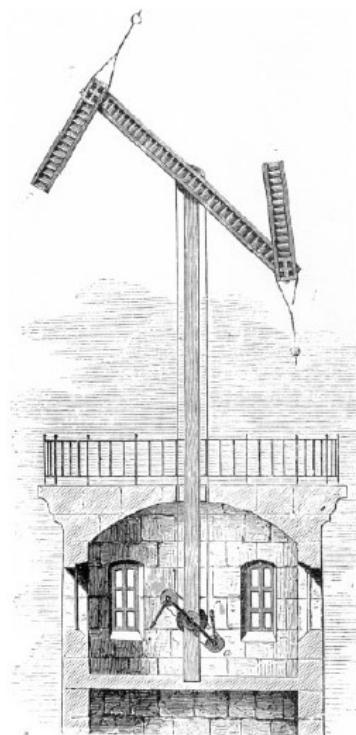


Figura 2 – Telégrafo Óptico de Claude Chappe.
Fonte: HUURDEMAN, 2003.

Todas as técnicas de comunicação óptica apresentadas até aqui, porém, podem ser vistas como formas variadas de comunicação visual, já que ainda se fazia necessária a percepção e interpretação humana para a captação e decodificação da mensagem transmitida. Em 1880, porém, Graham Bell e seu assistente Sumner Tainter apresentaram um instrumento de telefonia óptica que era capaz de transmitir a luz do sol modulada pelo som. Esse instrumento foi chamado por eles de fotofone (HUTT *et al.*, 1993).

No transmissor do aparelho (Figura 3), um espelho refletia a luz do sol concentrando-a, com o auxílio de lentes, em um diafragma localizado próximo ao bocal. Ao diafragma era afixada uma placa fina de vidro com revestimento de prata que, ao ser movimentada pela vibração do som, fazia variar a intensidade de luz emitida. O receptor era um espelho parabólico que concentrava o raio de luz em seu foco, onde era afixada uma célula de selênio. Como a resistência da célula de selênio varia de acordo com a intensidade de luz, ao ser conectada em série com uma bateria e fones de ouvido, tornava-se possível ouvir o som original transmitido (HUTT *et al.*, 1993).

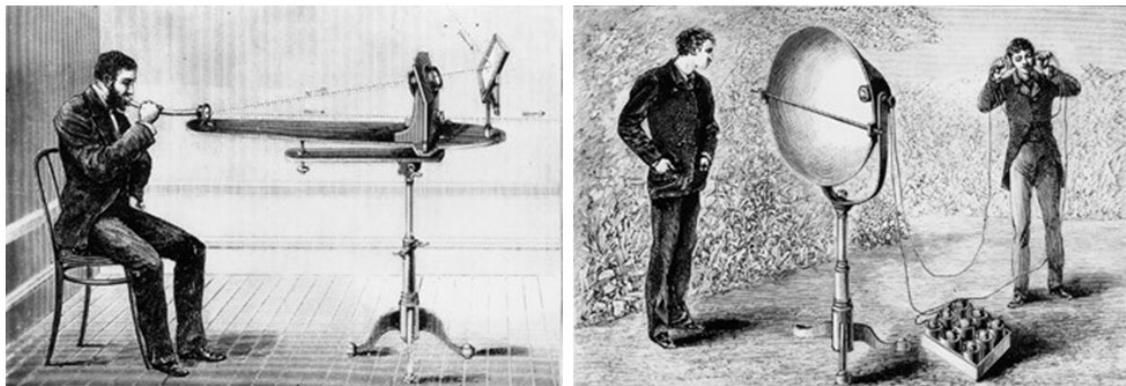


Figura 3 – Módulo transmissor (esquerda) e receptor (direita) do fotofone de Graham Bell.
Fonte: Adaptado de OSA, 2017.

Apesar da invenção de Bell, por muitas décadas os avanços em telegrafia elétrica, telefonia e rádio transmissão dominaram a área de telecomunicações. Com a invenção do laser - por Theodore Maiman em 1959 - e do trabalho com fibras ópticas de material extremamente puro - de Charles K. Kao e George A. Hockham em 1966 – despertou-se um novo interesse por uma forma de comunicação óptica que permitiria altas taxas de transmissão através de luz guiada em meio confinado. Tecnologia essa bastante utilizada nos dias de hoje (HUURDEMAN, 2003).

Em 1979, Fritz R. Gfeller e Urs Bapst publicaram seu trabalho sobre o uso de transmissões difusas na região do espectro do infravermelho (IR) para comunicação óptica doméstica, utilizando LEDs emissores e fotodiodos como receptores, mostrando a capacidade de se atingir até 1 Mbps de taxa de transmissão. O trabalho de Gfeller e Bapst foi fundamental para o desenvolvimento dos sistemas modernos de VLC.

2.2 VLC e Li-Fi

O espectro de radiação eletromagnética (Figura 4) compreende as frequências de radiação eletromagnética que se estendem desde as ondas de rádio (ondas longas, baixa frequência) até a radiação gama (ondas curtas, alta frequência). A banda ISM (*Industrial, Scientific and Medical*) consiste de regiões do espectro definidas pela ITU-T (ITU *Telecommunication Standardization Sector*) que correspondem às frequências reservadas para o desenvolvimento industrial, científico e médico sem a necessidade de licenciamento para utilização das mesmas. No Brasil, três faixas são reservadas para a banda ISM: 902 – 928 MHz, 2,4 – 2,4835 GHz e 5,725 – 5,875 GHz.

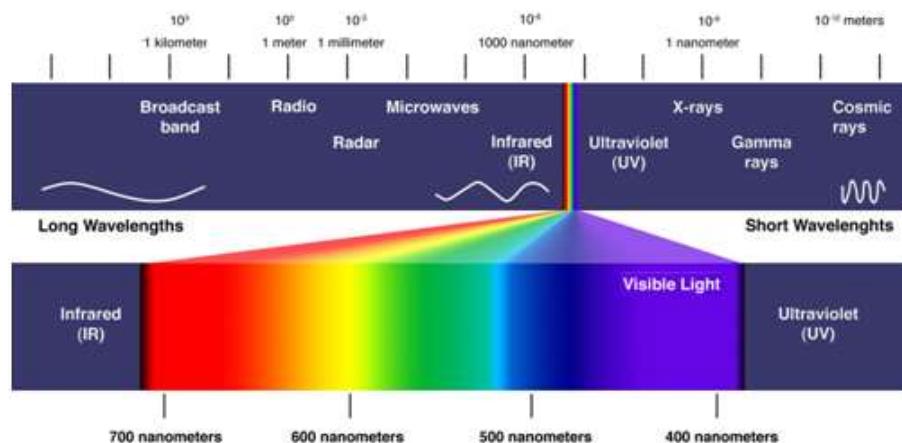


Figura 4 – Espectro de radiação eletromagnética.

Fonte: DAILYOPTICIAN, 2015.

A maioria dos dispositivos eletrônicos e tecnologias sem fio de uso pessoal operam dentro dessas faixas de frequência: Wi-Fi (celular, tablet, notebook), Bluetooth, telefone, mouse, teclado, fone de ouvido, impressora, brinquedos, entre outros. Dimitrov e Haas (2015) apontam que, devido ao aumento exponencial da demanda por

comunicação de dados sem fio por dispositivos móveis, a eficiência espectral da rede tende à saturação, apesar dos novos padrões e tecnologias desenvolvidos na área, acarretando perda da qualidade do sinal causada pela interferência com outros dispositivos que operam na mesma frequência.

Portanto, o VLC tem sido visto como uma potencial solução para o problema do esgotamento da banda ISM, já que, por operar na faixa de frequência de luz visível (400nm a 700nm), os dispositivos que utilizarem VLC não estarão sujeitos à massiva interferência dos modems e roteadores Wi-Fi, que trabalham na faixa de micro-ondas.

Um sistema VLC genérico, como o apresentado na Figura 5, utiliza LEDs para transmitir dados sem fio pela modulação de intensidade (IM) de luz visível e, no receptor, um fotodiodo (PD) para detectar o sinal por detecção direta (DD), técnica comumente utilizada em sistemas com fonte de luz incoerente por ser simples e de baixo custo (NEZAMALHOSSEINI *et al.*, 2013). O VLC foi criado como uma técnica de comunicação ponto-a-ponto, essencialmente um substituto do par metálico. Com a implementação de novas técnicas de transmissão, como *Direct-Current Biased Optical OFDM* (DCO-OFDM) e MIMO (*Multiple-Input Multiple-Output*), proporcionou-se o desenvolvimento da tecnologia que já mostrou alcançar taxas de até 3 Gbps. Esforços têm sido feitos para a especificação do VLC como parte do padrão IEEE 802.15.7 que, mais recentemente, tem sido revisto para a inclusão do Li-Fi.

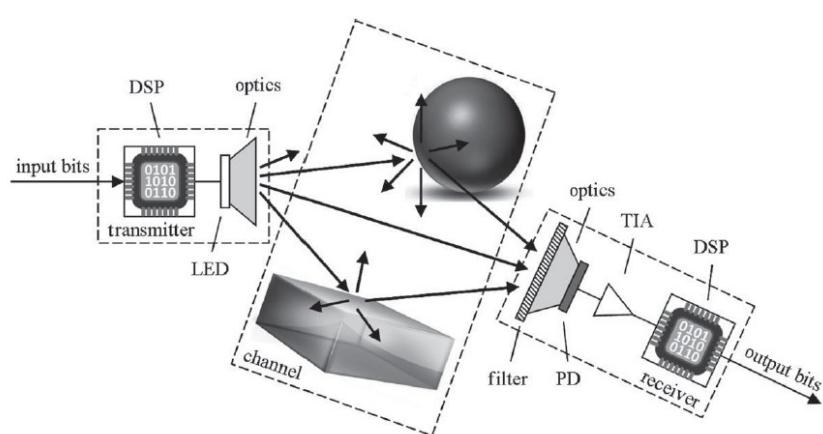


Figura 5 – Sistema VLC genérico, com blocos transmissor e receptor apresentados.
Fonte: DIMITROV & HAAS, 2015.

O termo Li-Fi ou *Light Fidelity* - em alusão ao Wi-Fi (*Wireless Fidelity*) - foi usado pela primeira vez pelo físico alemão Harald Haas em 2011, em sua palestra no TED Global sobre VLC. A ideia básica por trás dessa tecnologia é a de que os dados podem ser transmitidos pela luz de um LED, com sua intensidade variando mais rápido do que a percepção do olho humano é capaz de captar. “Dados através da iluminação”, como referido por Haas, seria uma forma de comunicação óptica por luz visível que, além de transmitir dados, também iluminaria o ambiente (SHARMA *et al.*, 2014).

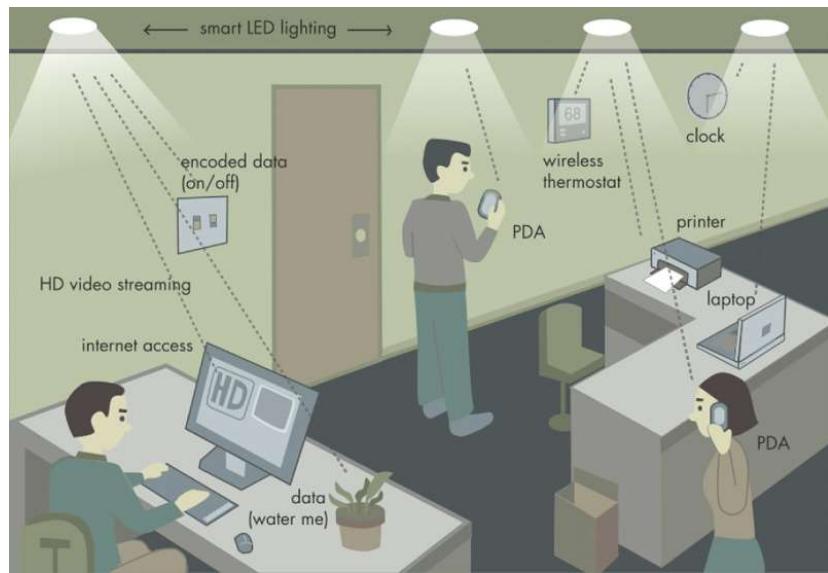


Figura 6 – Cenário de uso da tecnologia Li-Fi.

Fonte: JOHNSON, 2010.

De forma simplificada, a tecnologia VLC ou Li-Fi pode ser explicada da seguinte forma: quando o LED está aceso, é transmitido o símbolo binário “1”. Quando apagado, transmite-se o símbolo binário “0”. Chaveando-se eletronicamente a iluminação de forma acelerada, torna-se possível a transmissão sem fio de uma grande quantidade de dados em um curto intervalo de tempo.

Embora os conceitos de VLC e Li-Fi estejam intimamente interligados, ambos apresentam níveis distintos de descrição. Enquanto que o VLC está mais concentrado na definição do link físico óptico do sistema, o Li-Fi se mostra muito mais como um sistema completo de rede sem fio (Figura 7). Além de englobar as características do VLC, como *front-end* óptico e subsistemas - que controlam, por exemplo, modulação do sinal óptico e conversão analógica-digital – também estima um modelo de canal preciso que leva em

consideração a composição espectral do sinal. Ainda utiliza algoritmos para maximizar o *throughput* de dados, através da otimização da modulação (HAAS *et al.*, 2015).

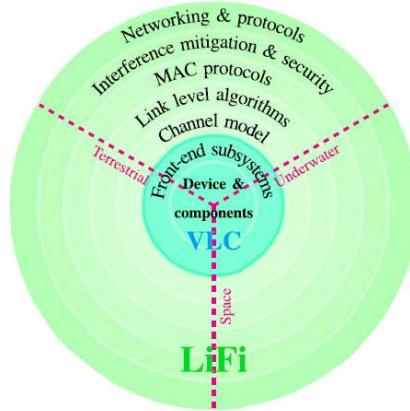


Figura 7 - Principais camadas do Li-Fi e VLC e áreas de aplicações.
Fonte: HAAS *et al.*, 2015.

Como dito anteriormente, o VLC foi concebido como um sistema ponto-a-ponto. O Li-Fi, pelo contrário, inclui comunicação bidirecional multiusuário, isto é, ponto-a-multiponto ou multiponto-a-ponto. Para habilitar o acesso multiusuário, define-se uma camada de protocolo de MAC que considera as especificações da camada física óptica. Da mesma forma, técnicas de eliminação de interferência são utilizadas para garantir a integridade dos dados recebidos e o *throughput* do sistema, além do uso de protocolos de redes que compatibilizam o sistema com as redes atuais existentes.

2.3 Li-Fi vs. Wi-Fi

A tecnologia Wi-Fi se faz bastante presente nos dias atuais, sendo a principal forma de acesso móvel à internet. Por operar na banda ISM, nas faixas de 2,4GHz e 5GHz, a tecnologia terá que enfrentar o futuro congestionamento dessa região do espectro eletromagnético, que resultará na elevada interferência e perda de *throughput*. Com o advento do Li-Fi, surgem novas possibilidades de aplicação e de se evitar essa competição por largura de banda, já que a tecnologia opera na região do espectro visível.

Na Figura 8, são comparados dois sistemas: um sistema típico de uma rede local sem fio utilizando Wi-Fi; e um exemplo de um sistema Li-Fi. No primeiro, um roteador Wi-Fi é conectado ao ponto de rede por um cabo ethernet e disponibiliza pela sua rede local sem fio o acesso à rede ao usuário. No segundo, utiliza-se um BPL (*Broadband Over Power Line*) para disponibilizar banda larga através da rede elétrica. A luz da lâmpada LED é então modulada de forma a transmitir o sinal ao receptor apropriado presente no computador do usuário.

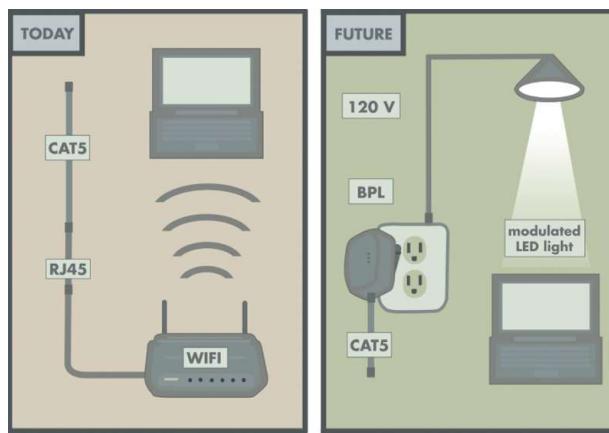


Figura 8 - Comparação entre acesso à internet via Wi-Fi e via Li-Fi.
Fonte: JOHNSON, 2010.

Abaixo são listadas algumas vantagens que o Li-Fi apresenta em comparação ao Wi-Fi ou às redes móveis 3G e 4G:

a) Capacidade

O Li-Fi pode alcançar taxas de transmissão muito mais altas do que atualmente o Wi-Fi e tecnologias similares oferecem, justamente por ter uma largura de banda muito maior. Estudos recentes mostraram que é possível atingir uma taxa de 3 Gbps utilizando um único micro-LED, e mais de 100 Gbps com laser LEDs (LDs) combinados a um difusor óptico para conseguir iluminação ampla (HAAS *et al.*, 2015). A Tabela 1 lista as taxas já alcançadas pelas principais tecnologias que operam na faixa do Wi-Fi em comparação com o Li-Fi.

Tabela 1 - Taxas de dados alcançáveis pelas tecnologias Wi-Fi em comparação com o Li-Fi.
Fonte: Adaptado de WIKIPEDIA, 2017.

Tecnologia	Frequência	Taxa alcançável
Wi-Fi IEEE 802.11n	2,4 / 5 GHz	150 a 600 Mbps
Wi-Fi IEEE 802.11ac	5 GHz	1300 Mbps
WiGig IEEE 802.11ad	2,4 / 5 / 60 GHz	7 Gbps
Li-Fi	430 a 750 THz	>100 Gbps

Além disso, como já dito anteriormente, o Wi-Fi opera com ondas de rádio, cujo espectro tem se tornado saturado com o uso crescente de dispositivos que operam nessa região. Por trabalhar na região do espectro visível, o Li-Fi sofre muito menos interferência de outros dispositivos, o que contribui para que o sistema alcance taxas mais próximas do ideal.

b) Eficiência

Com a popularização e uso de lâmpadas LED em casas, escritórios e na iluminação pública, devido à eficiência energética dos LEDs, a implantação de uma rede celular Li-Fi sobre a infraestrutura existente se torna viável e surge como um possível futuro benefício da tecnologia. Funcionando com células de tamanho reduzido, que leva ao conceito das Li-Fi *attocells*, essa nova rede móvel pode vir a ser uma das tecnologias que irão compor as redes sem fio heterogêneas da 5ª Geração da internet móvel.

O uso do Li-Fi nas redes celulares representaria um grande avanço, já que o enorme consumo das atuais ERBs (Estações Rádio Base) – boa parte devido mais para o resfriamento do sistema do que para a transmissão em si – poderia ser显著mente reduzido, tornando o sistema mais eficiente do ponto de vista energético.

c) Disponibilidade

A restrição do uso de celulares em certos ambientes, como aviões e hospitais, devido à interferência causada pelas ondas de rádio nos aparelhos de orientação e comunicação ou equipamentos hospitalares, é um fator que limita a disponibilidade do serviço aos usuários. Com a utilização de lâmpadas LEDs para a transmissão de dados, essa interferência é inexistente e, portanto, a disponibilidade do serviço é garantida.

d) Segurança

As ondas de rádio do Wi-Fi, devido ao seu comprimento de onda da ordem de milímetros, podem penetrar por paredes, o que se por um lado permite uma cobertura maior do sinal, também permite que este seja interceptado e, caso sua encriptação seja quebrada, pode comprometer a segurança do sistema.

O espectro de luz visível, no entanto, não é capaz de penetrar paredes, o que torna um sistema Li-Fi mais seguro, já que não é possível interceptar um sinal que esteja confinado em um ambiente fechado. As possíveis aplicações incluem a cobertura de instalações militares, bancos ou até mesmo a manutenção do sigilo de informações de empresas e laboratórios de pesquisa.

No entanto, apesar das vantagens, a tecnologia apresenta ainda algumas limitações. A intensidade luminosa incidente no sensor é um fator a ser observado, pois, dependendo do grau, pode levá-lo à saturação e piorar a qualidade do sinal. Em casos de aplicações em ambiente externos, por exemplo, a incidência de luz solar direta é um empecilho, já que pode reduzir bastante a taxa ou até mesmo inviabilizar a comunicação.

O fato da luz não ser capaz de penetrar paredes, apesar de ser uma vantagem do sistema no que diz respeito à segurança do mesmo, também pode ser uma desvantagem em determinadas aplicações, como por exemplo, quando se deseja uma cobertura de

sinal que atenda diversos ambientes isolados. Nesse caso, um sinal Wi-Fi torna-se mais apropriado, devido à boa penetrabilidade das ondas de rádio em obstáculos opacos.

Ainda, para um sistema de comunicação Li-Fi completo, é preciso que este seja full-duplex, com conexão *uplink* dos terminais móveis até o ponto de acesso óptico. Para isso, técnicas como TDD (*time-division duplexing*) e FDD (*frequency-division duplexing*) são cogitadas, porém enquanto na primeira é necessário garantir uma sincronização extremamente precisa para que as transmissões *uplink* e *downlink* sejam intercaladas no tempo, a segunda se mostra inviável, já que sistemas super-heteródinos não podem ser usados em sistemas do tipo IM/DD, como o Li-Fi. A solução mais viável, portanto, é a separação dos canais de *downlink* e *uplink* por comprimento de onda (WDD – *wavelength-division duplexing*). Nesse caso, o canal de *uplink* poderia ser estabelecido via transmissão infravermelha, por exemplo.

Portanto, o Li-Fi não deve ser visto como um substituto do Wi-Fi e de outras tecnologias sem fio, pois, mesmo com suas grandes vantagens em relação à taxa de dados, segurança e eficiência, ainda não é um sistema que atende a todos os tipos de aplicações. Contudo, vê-lo como uma tecnologia complementar às já existentes é essencial para a construção de uma rede diversificada que atenda as várias necessidades dos usuários, garantindo a mobilidade e a velocidade de conexão. Como uma das principais apostas para o 5G, no qual uma das ideias chaves é a separação de cenários *outdoor* e *indoor* (WANG *et al.*, 2014), o Li-Fi tem papel fundamental, já que tem potencial de atender ambientes *indoor* com alto grau de desempenho. Como tecnologia complementar, o Li-Fi também contribui para a composição de uma arquitetura heterogênea, com *macrocells*, *microcells* e *small cells*, aumentando a capacidade do sistema e auxiliando no cumprimento dos requisitos de performance do futuro 5G.

Capítulo 3

Projeto

3. Projeto

O protótipo de sistema VLC proposto foi desenvolvido de forma a transmitir dados unidirecionalmente de um ponto a outro através do ar pela variação de intensidade luminosa. O esquema simplificado do sistema pode ser visto na Figura 9 abaixo.

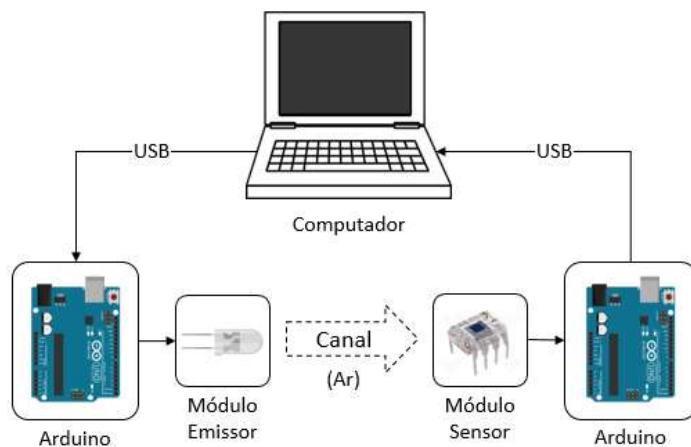


Figura 9 - Esquema simplificado de montagem do protótipo de sistema VLC.
Fonte: Elaboração própria.

A mensagem é processada pelo computador, utilizando o software MATLAB, e enviada ao microprocessador ATmega328, presente na plataforma de prototipagem Arduino Uno. O microprocessador controla a intensidade do LED de acordo com os dados recebidos do computador. O transmissor é exposto ao ambiente *indoor*, tendo o ar como canal de comunicação. Na recepção, um sensor detecta a variação de intensidade luminosa e a converte para sinal elétrico, transmitindo para outro microprocessador ATmega328. Este, por sua vez, envia os valores recebidos novamente para o computador (ou para outro computador com o mesmo sistema) que faz o devido processamento e apresenta os dados recebidos para o usuário.

Os seguintes componentes foram utilizados para a montagem do protótipo:

- Um computador;
- 2 x Arduino Uno (ATmega328);
- Driver LED (ver Seção 3.3.2);
- Sensor de Luminosidade.

Dois sistemas foram utilizados para o projeto para fins de comparação: OOK (*On-Off Keying*) e DCO-OFDM (*DC-Biased Optical OFDM*). No decorrer deste capítulo, serão abordados com maiores detalhes cada elemento que compõe o sistema, incluindo as técnicas de comunicação, circuitos e componentes utilizados.

3.1 OOK (*On-Off Keying*)

A codificação de linha por chaveamento *On-Off* (OOK) é uma técnica simples que representa dados binários pela presença ou ausência da onda portadora sendo, portanto, uma técnica de transmissão de portadora única (*single-carrier*). A transmissão de dados nas comunicações ópticas com fontes de luz incoerentes é realizada através da modulação de intensidade e detecção direta (IM/DD).

O chaveamento *On-Off*, ou também chamado de 2-PAM (*Pulse-Amplitude Modulation*), é uma das técnicas de modulação que podem ser aplicadas a esses tipos de comunicação, por permitir valores de tensão reais e não-negativos. Na figura 10, é mostrado um exemplo de transmissão modulada em OOK, onde a corrente do sinal é alternada entre dois níveis diferentes em torno de um nível DC, de acordo com os dados binários que estão sendo transmitidos.

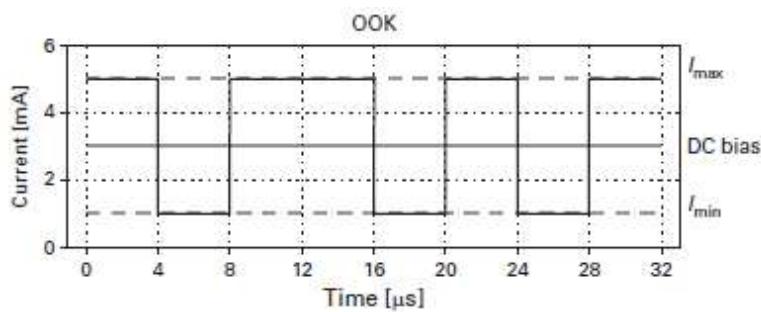


Figura 10 - Transmissão Óptica OOK.
Fonte: DIMITROV & HAAS, 2015.

No transmissor do sistema idealizado para esse projeto (Figura 11), a mensagem a ser enviada é codificada em símbolos binários. O conversor digital-analógico modula a intensidade de luz do LED de acordo com os símbolos binários recebidos. O símbolo binário “0” é representado pela ausência de luz (ou nível de tensão mais baixo) e o símbolo binário “1” é representado pela presença de luz (ou nível de tensão mais alto).

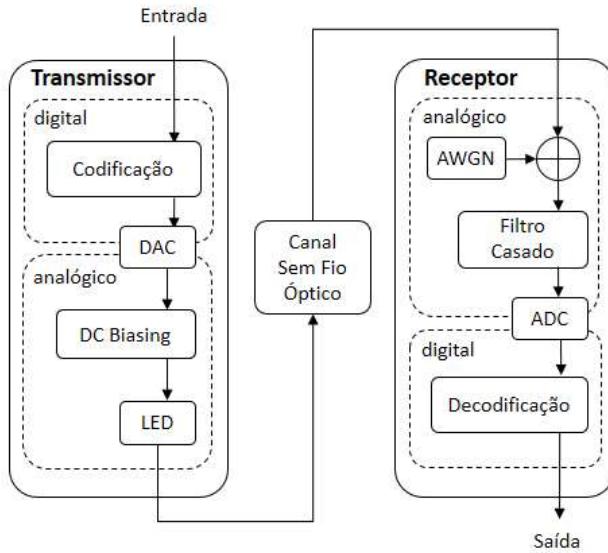


Figura 11 - Diagrama de blocos de transmissão óptica usando OOK.
Fonte: Adaptado de DIMITROV & HAAS, 2015.

O sinal luminoso passa pelo canal óptico e é detectado pelo receptor junto ao ruído do canal. Após a conversão de analógico para digital do sinal recebido, é realizada a decodificação para retornar à mensagem original. Normalmente, os sistemas OOK utilizam algum método de equalização, como *Zero-Forcing* ou MMSE, para compensar os efeitos causados pelo canal. Porém, no sistema desse projeto não será utilizado (ANEXOS F a I), ficando para ser implementado apenas no sistema OFDM, que será explicado na próxima seção.

É também importante ressaltar que sistemas OOK são suscetíveis à distorção em baixa frequência devido a flutuações no nível DC dos componentes elétricos, como também à oscilação das fontes de luz de fundo, que podem degradar o SNR (*Signal-to-Noise Ratio*) e prejudicar a recuperação da mensagem enviada (DIMITROV & HAAS, 2015).

3.2 DCO-OFDM (*DC-Biased Optical OFDM*)

Conforme aumenta-se a velocidade de transmissão em sistemas como OOK, PAM, PPM ou PWM, aumenta-se também a interferência entre símbolos (ISI) causada pela dispersão em frequência do canal óptico. O efeito dessa interferência degrada o sinal e reduz o SNR, sendo necessárias técnicas de modulação mais robustas à interferência, como é o caso do OFDM (*Orthogonal Frequency-Division Multiplexing*).

A técnica OFDM aplicada a sistemas SISO (*Single-Input Single-Output*) consiste na divisão da mensagem a ser enviada em blocos de dados, cujas entradas (símbolos) modulam uma subportadora, ou tom. Diferentemente da técnica OOK, portanto, OFDM é uma forma de transmissão multiportadora (*multi-carrier*).

Para facilitar a recuperação dos símbolos no receptor, as subportadoras devem ser ortogonais entre si, de forma que na frequência central de cada subportadora, todas as outras tem amplitude igual a zero no domínio da frequência (Figura 12). Assim, transmissões OFDM ocupam o espectro de frequência de forma mais eficiente do que transmissões FDM, que necessitam manter as frequências centrais das subportadoras mais afastadas entre si para evitar sobreposição (DINIZ *et al.*, 2012).

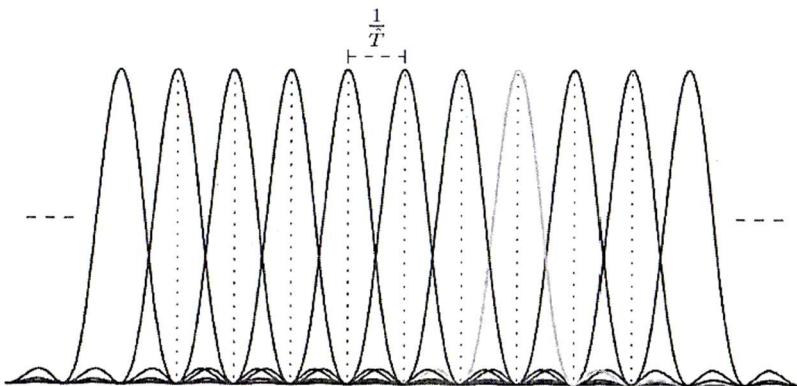


Figura 12 - Representação das subportadoras OFDM no domínio da frequência.
Fonte: DINIZ *et al.*, 2012.

Segundo Diniz *et al.* (2012), a interferência entre blocos (IBI), ou interferência entre símbolos OFDM, causada pelas versões atrasadas e atenuadas do sinal transmitido em um canal com multipercursos, dificulta a manutenção da ortogonalidade das subportadoras quando essas chegam no receptor. Para evitar esse tipo de interferência é utilizado um período de guarda, que funciona como um símbolo OFDM estendido.

Uma das técnicas que fazem uso desse tipo de extensão, e que será usada nesse projeto, insere no período de guarda uma cópia de uma parte inicial do símbolo OFDM (Figura 13). Essa extensão é chamada de prefixo cíclico. A introdução dessa redundância diminui o *throughput* do sistema, mas consegue manter a ortogonalidade, desde que o prefixo cíclico seja descartado no receptor, eliminando assim a IBI. A interferência remanescente (ISI) é eliminada no receptor usando a técnica de equalização de canal conhecida como *Zero-Forcing* (DINIZ *et al.*, 2012).

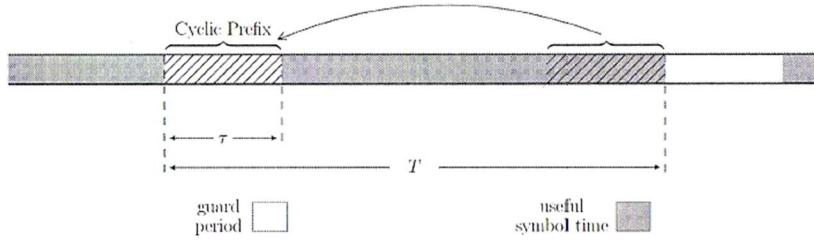


Figura 13 - Inserção do Prefixo Cíclico no período de guarda entre símbolos OFDM.
Fonte: DINIZ et al., 2012.

Considerando o sistema linear invariante no tempo da Figura 14, vemos que esse realiza a convolução linear do sinal de entrada $x(n)$ com a resposta ao impulso do sistema $h(n)$.

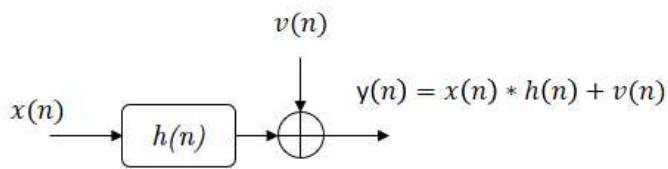


Figura 14 - Diagrama de Blocos de um sistema linear invariante no tempo.
Fonte: Elaboração própria.

Usando a técnica de Overlap-and-Save, podemos realizar a convolução dos blocos da sequência de entrada com o sistema, fazendo a inserção do prefixo cíclico. Se considerarmos o canal óptico $h(n)$ como um filtro FIR de tamanho L e o representarmos como uma matriz do tipo Toeplitz $H(z)$, podemos torná-la circulante realizando a transformação descrita na equação (1). O índice M é o tamanho do bloco de entrada da sequência a ser enviada.

$$C = \begin{bmatrix} h(0) & \dots & h(L-1) & 0 & \dots & 0 \\ 0 & \dots & h(L-2) & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & h(0) & \dots & h(L-1) \end{bmatrix} \begin{bmatrix} I_M \\ \vdots \\ I_{L-1} \\ 0 \end{bmatrix} \quad (1)$$

$$C = \begin{bmatrix} h(0) & h(1) & \dots & h(L-1) & 0 & \dots & 0 \\ 0 & h(0) & h(1) & \dots & h(L-1) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & h(L-1) \\ h(L-1) & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & h(1) \\ h(1) & \dots & h(L-1) & 0 & 0 & \dots & h(0) \end{bmatrix} \quad (2)$$

Uma maneira equivalente de se obter a matriz circulante do canal é descrita em Sayed (2014). Nela, é realizada a transformação nos blocos de entrada, o que corresponde à inserção do prefixo cílico. Dessa forma, ao invés da sequência de dados s_n de tamanho M , será transmitida pelo canal a sequência com extensão cílica u_n de tamanho $M+L-1$.

$$u_n = \begin{bmatrix} I_M \\ \vdots \\ I_{L-1} \end{bmatrix} s_n = \begin{bmatrix} s_n \\ \vdots \\ s_n(0:L-2) \end{bmatrix} \quad (3)$$

Os sistemas OFDM utilizam blocos de matrizes DFT (F) e IDFT (F^*) tanto para modular/demodular, quanto para multiplexar/demultiplexar as subportadoras (Figura 15). Seu uso no sistema implica em uma convolução circular entre o sinal e o canal. Porém, transmitindo a sequência em blocos de tamanhos $N \geq M + L - 1$, é possível realizar uma convolução linear através de uma convolução circular (DINIZ *et al.*, 2014).

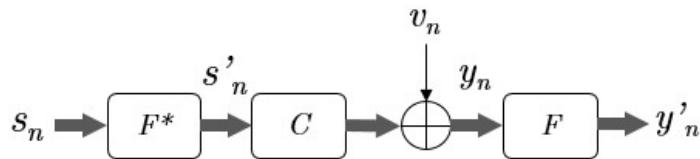


Figura 15 - Diagrama de Blocos OFDM simplificado.
Fonte: Elaboração própria.

$$y'_n = FCF^*s_n + Fv_n \quad (4)$$

Como qualquer matriz circulante C pode ser diagonalizada pela matriz DFT, podemos escrevê-la da seguinte forma (SAYED, 2014):

$$C = F^*\Delta F \quad (5)$$

Como $F^*F = FF^* = 1$ (onde F^* é o complexo conjugado de F) e substituindo (5) em (4), temos:

$$y'_n = FF^*\Delta FF^*s_n + Fv_n \quad (6)$$

$$y'_n = \Delta s_n + v'_n \quad (7)$$

Assim, considerando o ruído $v'_n = 0$, basta obter a estimativa da sequência enviada \hat{s}_n invertendo-se a matriz diagonal do canal Δ e multiplicando-a pela sequência recebida y'_n . Essa técnica de equalização é conhecida como *Zero-Forcing*, e com ela é possível eliminar a interferência entre símbolos (ISI) causada pelo canal.

$$\hat{s}_n = \Delta^{-1}y'_n \quad (8)$$

No entanto, é preciso primeiro obter a estimativa do canal Δ , já que na prática este não é conhecido. Um dos métodos possíveis de estimativa do canal é através da transmissão de uma sequência piloto conhecida junto com a mensagem. Essa sequência normalmente é enviada a intervalos de tempo, porém para este projeto será enviada apenas uma vez antes da mensagem.

Ao ser transmitida pelo canal, a sequência piloto sofre as mesmas condições que a mensagem e, ao chegar ao receptor, faz-se a correlação com a sequência que se esperava receber. Isso pode ser feito de forma similar à equação (7), apenas diagonalizando-se a sequência esperada s_n e multiplicando-a pelo vetor dos elementos da diagonal principal de Δ , que é uma matriz diagonal e, portanto, tornando a equação (9) e (7) equivalentes.

$$y'_n = S_n\lambda + v'_n \quad (9)$$

Considerando o ruído $v'_n = 0$ e invertendo-se S_n , temos:

$$\lambda = S_n^{-1}y'_n \quad (10)$$

Para obter Δ , basta diagonalizar λ , e então é possível realizar a equalização do resto da mensagem por (8), eliminando as interferências do canal.

Por fim, é importante lembrar que, diferentemente das comunicações RF, as comunicações ópticas permitem apenas valores reais e não-negativos. Por isso, é necessário garantir que na entrada do bloco DFT do transmissor a sequência a ser transmitida possua simetria hermitiana, para que a componente imaginária na saída do bloco DFT seja nula (SCHMIDT *et al.*, 2008).

Para isso, a entrada do bloco DFT deve receber a sequência s_n concatenada com sua versão complexa conjugada s_n^* espelhada. A Figura 16, assim como a equação (11), mostra o vetor de entrada s_n sendo mapeado para a entrada do bloco IFFT I_n . As entradas I_0 e $I_{M/2-1}$, que correspondem respectivamente às frequências DC e de Nyquist, são igualadas a zero, assim como a frequência de banda de guarda. Percebe-se que, com isso, há uma redução significativa no *throughput* do sistema, já que se aumenta a redundância.

$$I_0 \dots I_{M-1} = s_0 \dots s_{M/2-1}, s_{M/2}, s_{M/2-1}^* \dots s_1^* \quad (11)$$

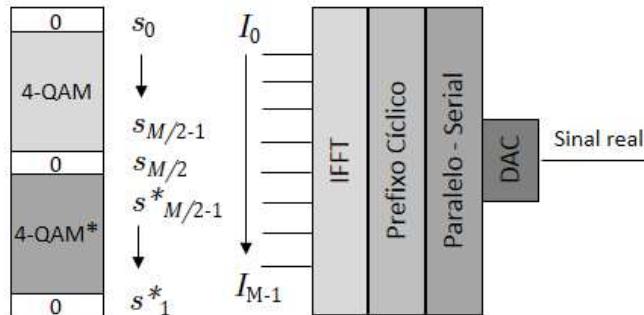


Figura 16 - Transmissor OFDM usando simetria hermitiana para comunicação óptica.

Fonte: Adaptado de SCHMIDT *et al.*, 2008.

Finalmente, na Figura 17 é apresentado o digrama completo do sistema OFDM que foi desenvolvido no MATLAB (ANEXOS A). Como a entrada de dados é serial, primeiramente é feita a conversão serial-paralelo, para a divisão da mensagem em

blocos de tamanho $M = 64$, e antes de enviar para o módulo transmissor, é realizada a conversão paralelo-serial. No receptor é feito o procedimento inverso.

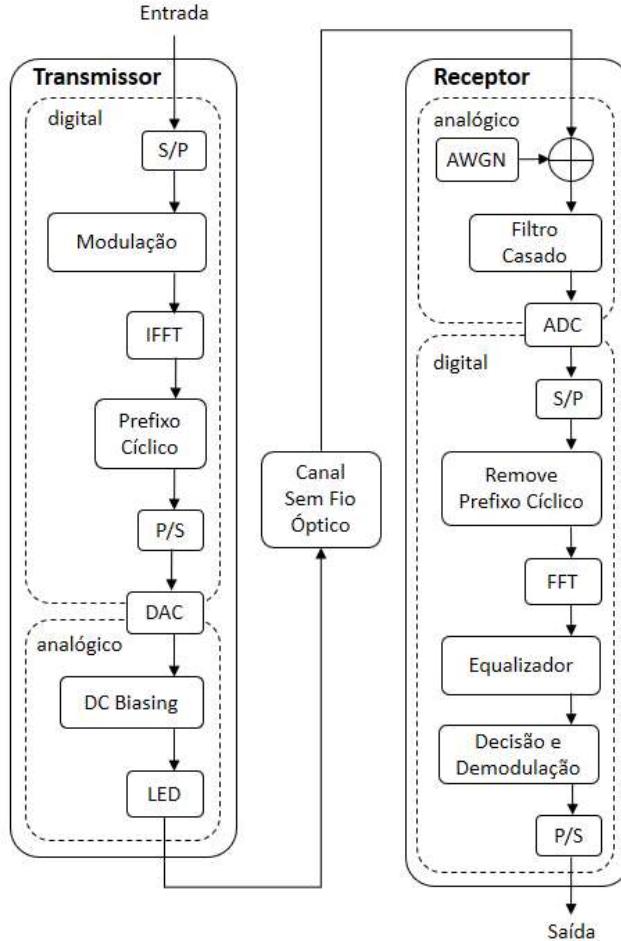


Figura 17 - Diagrama de blocos completo do sistema OFDM.
Fonte: Adaptado de DIMITROV & HAAS, 2015.

As técnicas de modulação interna utilizadas foram BPSK (*Binary Phase Shift Keying*) e 4-QAM (*Quadrature Amplitude Modulation*). Na primeira, os dados binários são modulados em dois símbolos (-1 e 1) defasados no plano complexo de 180°. Já na segunda técnica, a modulação ocorre com defasagem de 90° entre os quatro símbolos possíveis ($\pm \frac{\sqrt{2}}{2} \pm \frac{\sqrt{2}}{2} i$). A modulação BPSK é menos suscetível a erros na detecção causados pelo ruído, devido à distância maior entre os símbolos, porém transmite menos informação por símbolo do que a modulação em quadratura.

3.3 Transmissor Óptico

Para o módulo de transmissão óptica foi utilizado um microprocessador ATmega328, presente na plataforma de prototipagem Arduino Uno. Os blocos processados pelo computador no software MATLAB são enviados via interface serial ao Arduino, que usa uma de suas saídas analógicas para transmitir o sinal eletricamente até o LED.

Para o sistema DCO-OFDM, sabendo que as transmissões ópticas não permitem valores negativos, é realizado um *DC-biasing* para deslocar o nível DC do sinal e tornar positivo os valores negativos. No projeto, esse procedimento é feito tanto digitalmente (Figura 18) – já que as saídas analógicas do microprocessador não aceitam valores negativos – quanto analogicamente, o que será explicado na Seção 3.3.2.

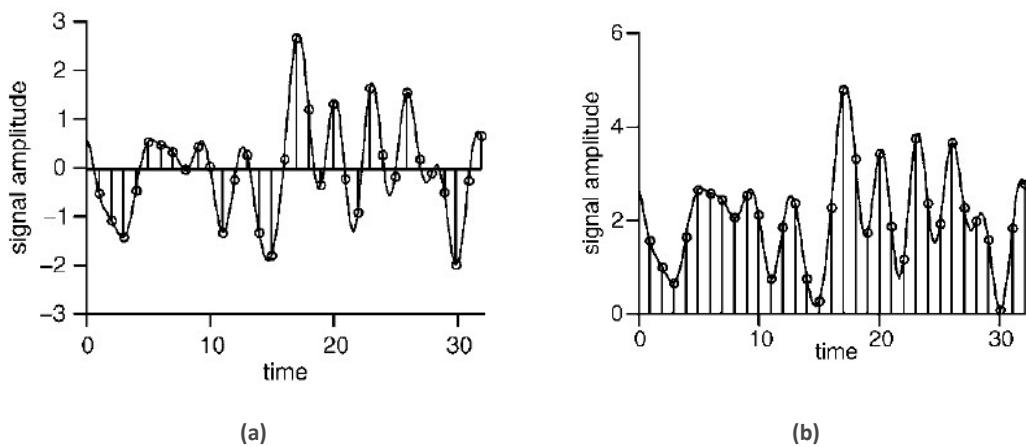


Figura 18 - Transmissão DCO-OFDM: (a) Sem DC-Bias (b) Com DC-Bias.
Fonte: Adaptado de ARMSTRONG & LOWERY, 2006.

3.3.1 DAC (Conversor Analógico-Digital)

As saídas analógicas realizam uma modulação por largura de pulso (PWM – *Pulse-Width modulation*) usando os valores recebidos como valor de porcentagem de *duty cycle*, que determinam o tempo que o sinal é mantido em nível alto (Figura 19). Esses devem assumir níveis de 0 a 255, já que a precisão do PWM é de 8 bits. No entanto, para obter o sinal de onda esperado, realizando a conversão digital-analógica (DAC), é necessário fazer a filtragem da frequência de PWM utilizando um circuito passa-baixas,

de forma a obter o nível médio do sinal. Assim, cada valor de *duty cycle* estará associado a um nível de tensão na saída do circuito. Para isso, foi usado um simples filtro RC passa-baixas de primeira ordem, como mostrado na Figura 20.

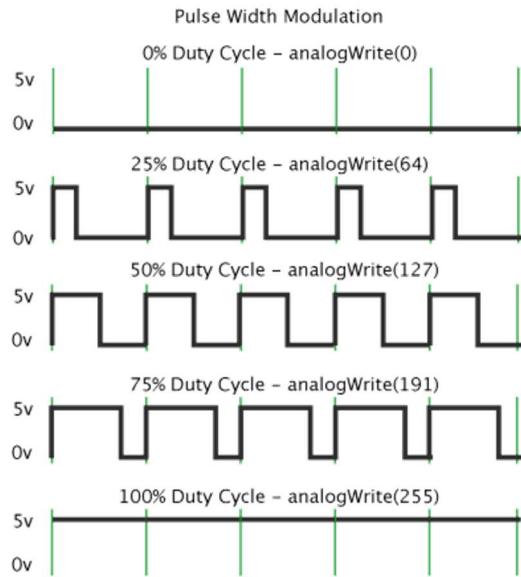


Figura 19 - Modulação PWM no ATmega328 com variação de *duty cycle*.
Fonte: ARDUINO, 2017.

A frequência de corte do Filtro RC Passa-Baixas do circuito pode ser calculada pela expressão abaixo:

$$f_c = \frac{1}{2\pi RC} \quad (12)$$

Os valores dos componentes foram dimensionados para $R = 1 \text{ k}\Omega$ e $C = 1 \mu\text{F}$. Substituindo-os em (12), temos o valor da frequência de corte $f_c = 159 \text{ Hz}$. O filtro, portanto, começa a atenuar a amplitude de frequências maiores que f_c , o que inclui a frequência PWM do ATmega328 de 31.372 kHz. Apesar dos testes realizados utilizarem frequências superiores, essas não são tão maiores que a frequência de corte, sendo pouco atenuadas. O uso desse valor de frequência de corte tem como objetivo principal eliminar ao máximo a variação causada pela frequência PWM, que prejudica a qualidade do sinal.

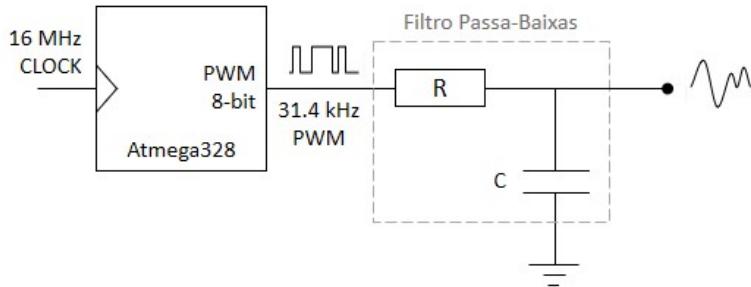


Figura 20 - DAC utilizando PWM e filtro RC passa-baixas.
Fonte: Elaboração própria.

3.3.2 Driver LED

Para o projeto, foi utilizado apenas um LED branco de alto-brilho 5mm com potência máxima de 100mW. A Figura 21 (a) mostra a distribuição espectral do LED, indicando um pico na região da luz violeta (~410nm), mas também na região da luz verde (~500nm). Na Figura 21 (b) é apresentado o diagrama de radiação do LED, que se mostra bastante diretivo, o que em aplicação *indoor* pode não ser ideal, já que quanto mais difundida é a iluminação, melhor é a captação ao redor do ambiente.

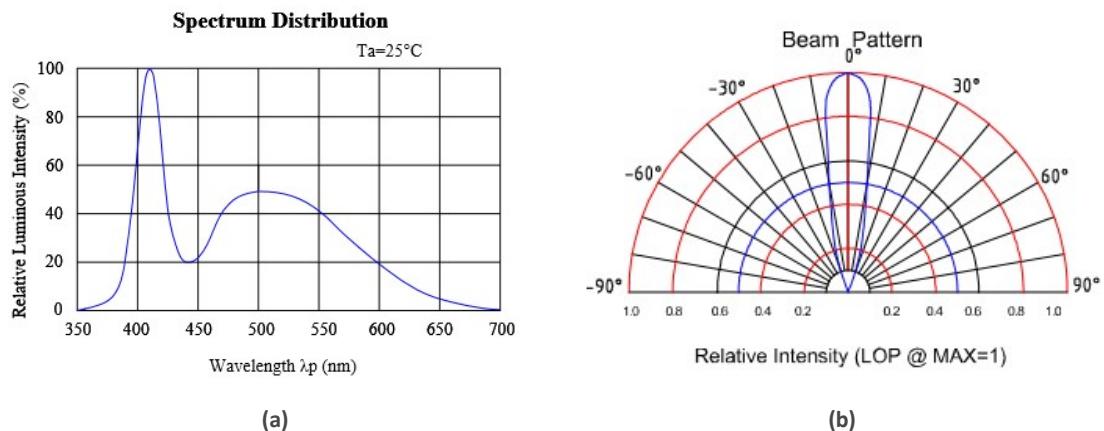


Figura 21 – Características do LED: (a) Distribuição Espectral (b) Diagrama de Radiação.
Fonte: LUCKYLIGHT, 2006.

Porém, no caso da utilização de vários LEDs, como uma lâmpada LED, torna-se necessário o uso de um *driver* para suportar a corrente consumida pelo conjunto. Assim, a corrente é drenada diretamente de uma fonte e o sinal a ser transmitido é controlado na base do transistor, cuja especificação da corrente de coletor está de acordo com a

corrente demandada. Por isso, foi adicionado ao circuito do transmissor o *driver* da Figura 22, utilizando os seguintes componentes:

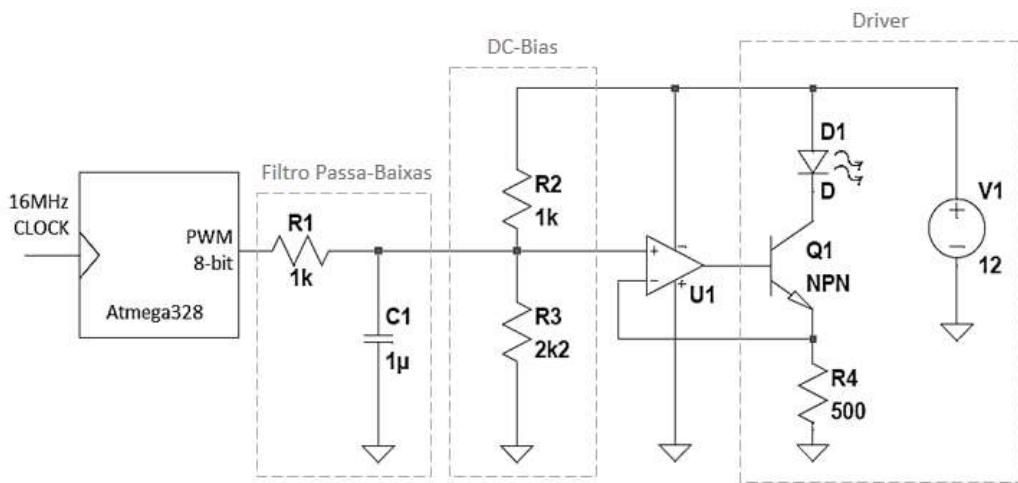


Figura 22 – Circuito completo do Transmissor Óptico com Driver.

Fonte: Elaboração própria.

- Resistores (R1, R2, R3 e R4);
- Capacitor (C1);
- Amplificador Operacional LM358 (U1);
- Transistor TIP31 NPN (Q1);
- Fonte 12VDC – 1A (V1);
- LED.

O amplificador operacional (U1) funciona como um *buffer*, isolando o Filtro RC do circuito do LED. Em seu pino de entrada é utilizado um divisor resistivo (R2 e R3) para fazer um *DC-biasing* analógico. Esse procedimento é necessário pois, devido à não-linearidade do LED e também do transistor (Q1), é preciso limitar o sinal a ser transmitido dentro da região com resposta mais linear do componente.

Outra forma de evitar a distorção causada pela não-linearidade dos semicondutores, mas que não está incluída neste projeto, é compensá-la realizando uma pré-distorção antes da conversão digital-analógica. Assim, é possível utilizar o máximo de excursão do sinal à saída do componente com pouca ou nenhuma distorção (DIMITROV & HAAS, 2015).

3.3.3 Algoritmo de Transmissão

O código do módulo transmissor foi escrito no *software IDE Arduino*, que é baseado na linguagem C, e pode ser visto com maiores detalhes no ANEXO B. O algoritmo da Figura 23 resume o procedimento de transmissão dos valores recebidos do computador pelo ATmega328.

O microprocessador possui limitações quanto à memória disponível para armazenamento de variáveis e, também, quanto ao tamanho da memória *buffer* da interface serial (64 bytes). Desse modo, não é possível enviar para o microprocessador toda a mensagem a ser enviada de uma única vez – no caso de uma grande quantidade de valores, como uma imagem, por exemplo. Por isso, a mensagem é enviada em blocos (de tamanho $M+L-1$, no caso do OFDM) e, ao término do envio de um bloco, esse é transmitido pela saída analógica, até chegar ao LED.

Além disso, é utilizada a transmissão de um valor de início de mensagem que corresponde ao nível mais baixo de tensão (0V), para que o receptor saiba quando começar a ler a mensagem. Essa é transmitida a uma taxa predeterminada, que varia de acordo com o teste que está sendo realizado.

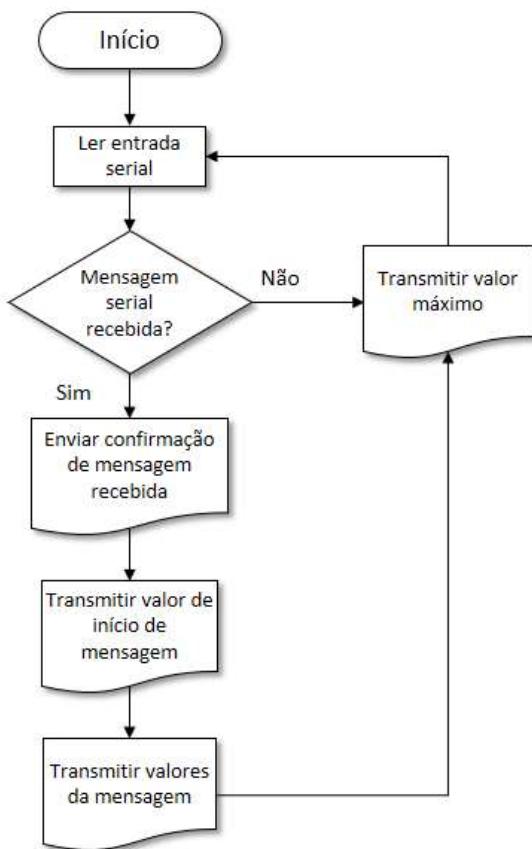


Figura 23 – Algoritmo do código de transmissão.
Fonte: Elaboração própria.

3.4 Receptor Óptico

Assim como o módulo de transmissão, foi utilizado para o módulo receptor um microprocessador ATmega328 (Arduino Uno). O sinal enviado pelo LED é detectado por um sensor de luminosidade ou fotodiodo, que faz a transdução do sinal óptico para sinal elétrico. Após amplificado e filtrado, esse sinal elétrico passa por um Conversor Analógico-Digital (ADC), presente no microprocessador. A precisão do conversor do ATmega328 é de 10 bits, possibilitando a representação de 0 a 5 Volts de entrada em 1024 níveis, com taxa de amostragem de 10 kHz. Os valores são amostrados novamente de acordo com a taxa de envio de símbolos do transmissor e os blocos recebidos são enviados via interface serial ao computador, para processamento no software MATLAB.

3.4.1 Sensor de Luminosidade

Para o sensor, foi escolhido o circuito integrado OPT101, que além do fotodiodo para converter o sinal luminoso em sinal elétrico, também possui um amplificador de transimpedância integrado, o que evita os efeitos de correntes de fuga e ruídos causados pelas capacitâncias parasitas, comuns nos circuitos de amplificação discretos (TEXAS INSTRUMENTS, 2015). Além disso, possui uma resposta com largura de banda de 14 kHz, suficiente para as taxas de transmissão utilizadas. Na Figura 24 é detalhado o circuito interno do CI e sua pinagem.

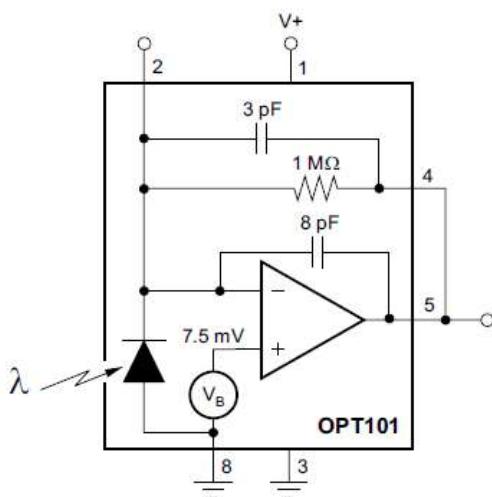


Figura 24 - Diagrama do circuito interno do CI OPT101.
Fonte: TEXAS INSTRUMENTS, 2015.

A resposta espectral do sensor, como pode ser vista na Figura 25 (a), tem seu pico na região do infravermelho (~ 850 nm), o que é comum na maioria dos sensores ópticos. Ainda assim, o sensor tem uma boa resposta na faixa da luz visível, que vai de 400 nm a 700 nm, aproximadamente. Na Figura 25 (b) pode ser observada a resposta do sensor relativa ao ângulo de incidência da luz. A resposta ao ângulo de incidência θ_x , relativa à vista frontal do sensor, apresenta um melhor ganho do que a resposta ao ângulo θ_y , relativa à vista lateral. Ambas, no entanto, possuem ganho igual e máximo quando a luz incide com ângulo de 0° no fotodiodo.

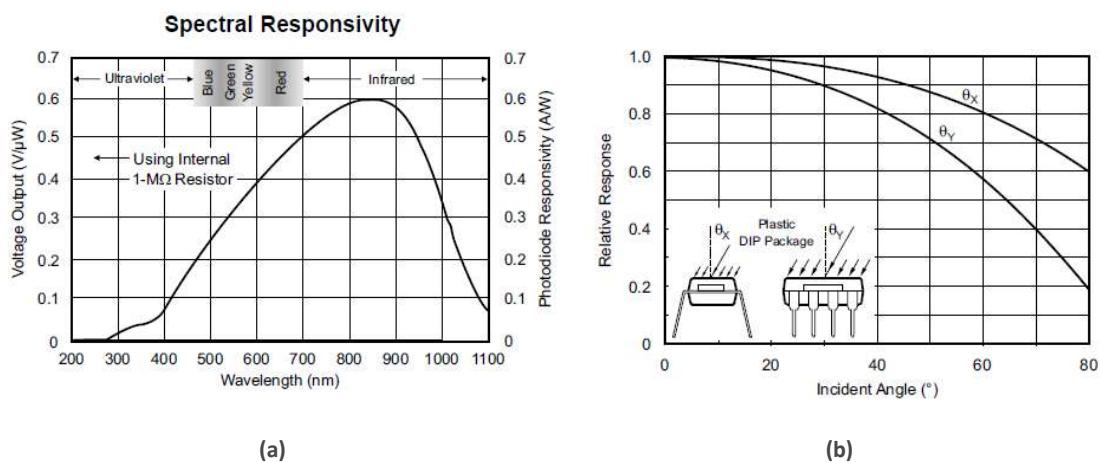


Figura 25 - OPT101: (a) Resposta Espectral (b) Resposta ao Ângulo de Incidência.
Fonte: TEXAS INSTRUMENTS, 2015.

3.4.2 Algoritmo de Recepção

A Figura 26 apresenta o algoritmo do código receptor, que resume os procedimentos seguidos pelo ATmega328 do módulo de recepção para o recebimento da mensagem enviada pelo transmissor óptico. O código completo pode ser visto no ANEXO B.

O microprocessador lê constantemente o sinal elétrico transmitido pelo sensor, à espera do valor de início de mensagem, que ocorre quando o nível amostrado do sinal assume um valor menor do que o limite pré-estabelecido, já que, quando não se está transmitindo mensagem, o nível do sinal é constante e o mais alto possível. Quando é

percebido o valor de início de mensagem, o módulo começa a amostrar o sinal na mesma taxa de transmissão do módulo transmissor.

Ao fim da leitura do bloco da mensagem, esse é enviado serialmente para o computador para ser armazenado até que todos os blocos da mensagem sejam recebidos e, em seguida, processados pelo código MATLAB.

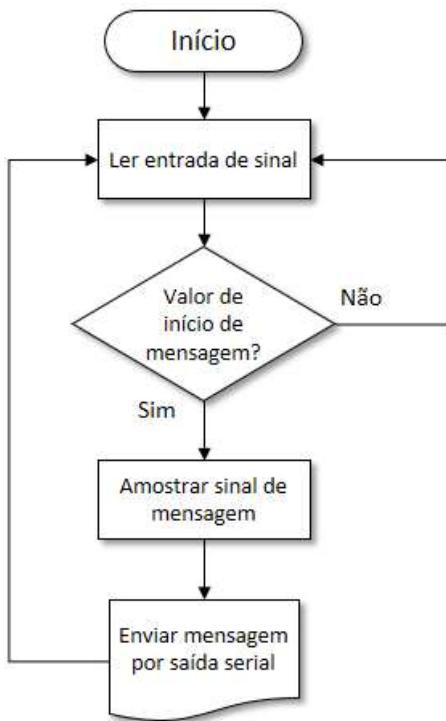


Figura 26 - Algoritmo do código de recepção.
Fonte: Elaboração própria.

Capítulo 4

Resultados e Discussão

4. Resultados e Discussão

Neste capítulo serão abordados os resultados obtidos, tanto na simulação do sistema óptico OFDM, quanto no experimento prático montado para os sistemas DCO-OFDM e OOK, com o objetivo de avaliar comparativamente o desempenho dos sistemas e verificar a viabilidade da implementação das técnicas desenvolvidas.

4.1 Simulação

Para a simulação do sistema óptico OFDM foi utilizado o código desenvolvido no software MATLAB (ANEXO C), que implementa digitalmente o sistema visto na Seção 3.2 deste documento.

As sequências de dados aleatórios foram transmitidas utilizando duas técnicas de modulação diferentes: BPSK e 4-QAM; com tamanho de 2400 e 4800 bits, respectivamente. A Figura 27 apresenta os resultados obtidos para uma transmissão com nível de SNR de 20 dB e canal de comprimento $L=41$. O tamanho M usado para os blocos de entrada foi de 64, de forma a ter um comprimento maior do que o comprimento do canal escolhido.

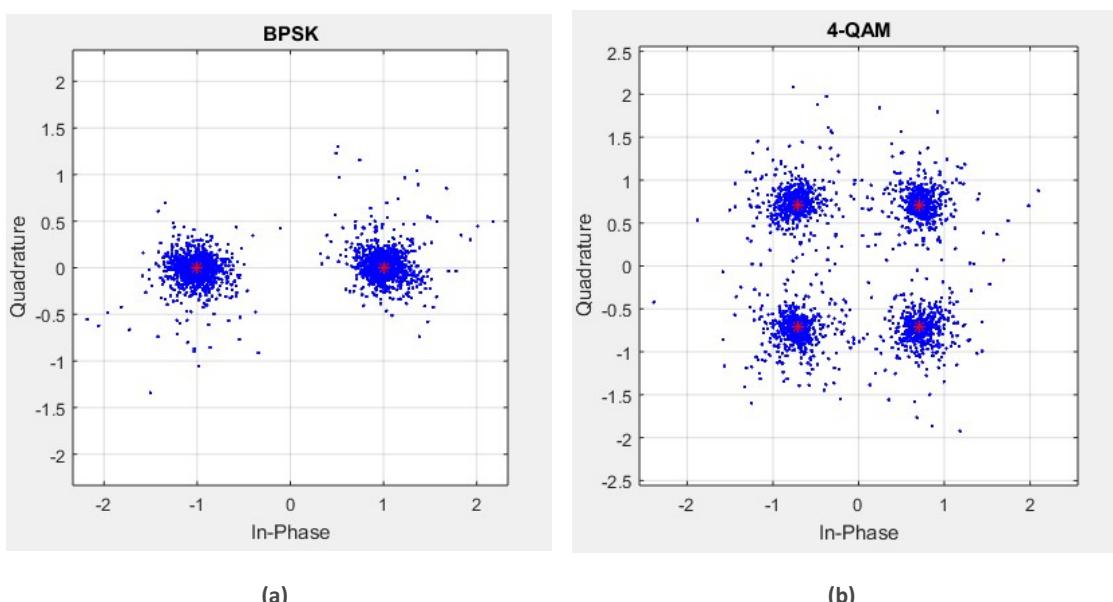


Figura 27 - Diagramas de constelação do sistema OFDM simulado: (a) BPSK (b) 4-QAM.

É possível notar pela Figura 27 que, ao utilizar a técnica de modulação BPSK, o processo de tomada de decisão tende a cometer menos erros, já que os símbolos estão mais afastados entre si. Enquanto que com a técnica 4-QAM, apesar de transmitir mais bits por símbolo – o que aumenta o *throughput* do sistema – mantém os símbolos mais próximos entre si, aumentando as chances de erros.

Para uma análise mais completa do desempenho do sistema em relação ao nível de ruído do canal, foi realizada uma varredura de valores de SNR, variando-os entre níveis de -20 dB a 20 dB com passo de 0,5 dB e montando um gráfico de BER (*Bit Error Rate*) x SNR.

Foi gerado um conjunto de 5000 canais aleatórios de comprimento $L=41$ para a realização de uma média entre os resultados de probabilidade de erro (BER) obtidos para cada SNR. A Figura 28 mostra o resultado dessa simulação, apresentando a curva de BER para as técnicas BPSK e 4-QAM.

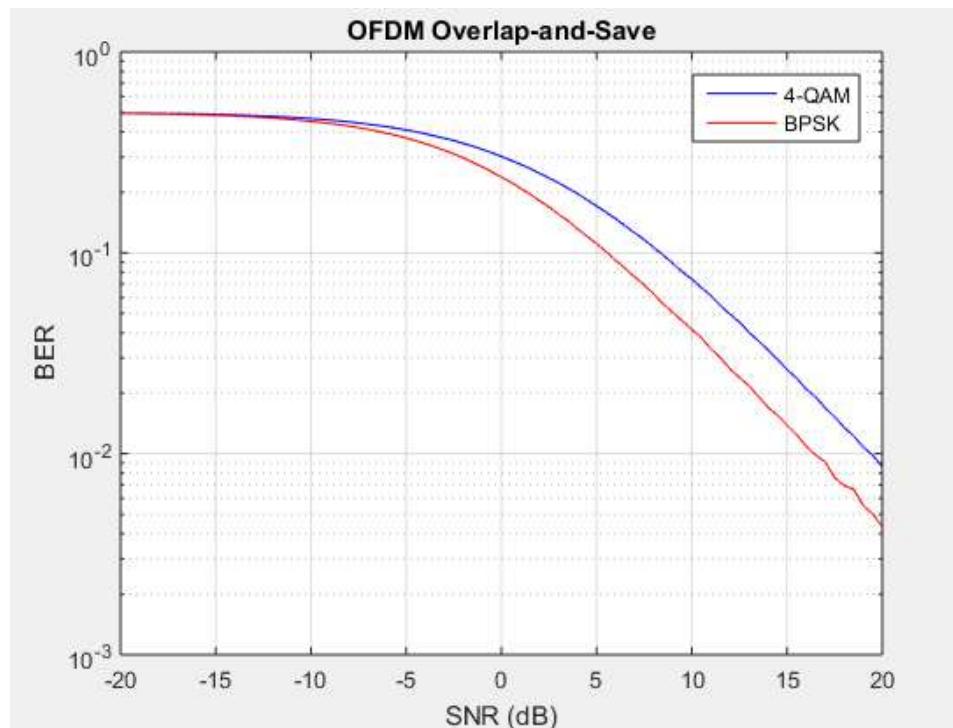


Figura 28 - Gráfico de BER x SNR para sistema OFDM.

Nota-se que o desempenho da técnica de modulação BPSK é, como esperado, melhor do que a modulação 4-QAM, possuindo uma curva mais acentuada, que decresce mais rapidamente conforme aumenta-se a relação sinal-ruído do canal (SNR).

4.2 Experimento Prático

A montagem dos circuitos dos módulos de transmissão e recepção do experimento prático foi realizada em *protoboards* pequenas, que se conectavam com as placas de Arduino através de *jumpers*. Para que o LED e o sensor estivessem em visada direta, suas *protoboards* foram coladas em suportes em formato de L, possibilitando assim um melhor alinhamento e a máxima qualidade possível de recepção do sinal (Figura 29).

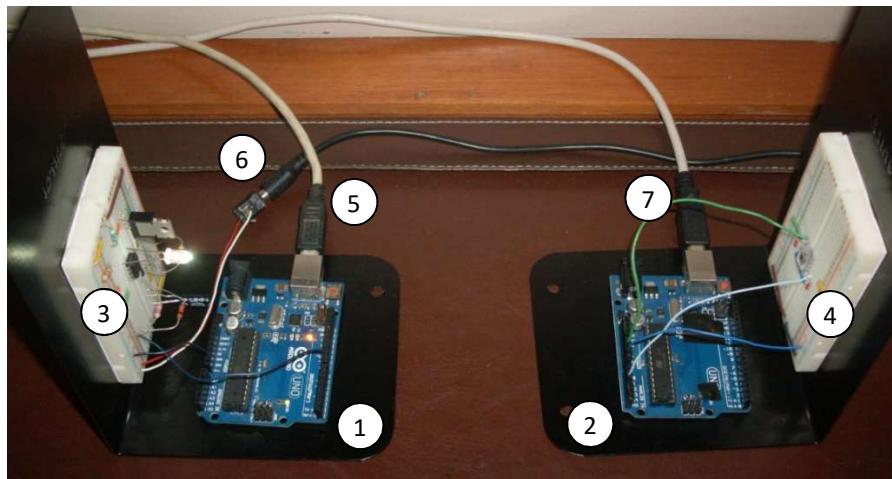


Figura 29 - Foto dos módulos transmissor (esquerda) e receptor (direita).

Os elementos do sistema da Figura 29 estão enumerados abaixo:

- 1) Placa Arduino Uno – Transmissor;
- 2) Placa Arduino Uno – Receptor;
- 3) Circuito transmissor (Filtro Passa-Baixas + Driver + LED);
- 4) Circuito receptor (CI OPT101 – Sensor de luminosidade);
- 5) Conector USB do módulo transmissor;
- 6) Cabo de alimentação 12VDC para Driver LED;
- 7) Conector USB do módulo receptor.

Os experimentos foram realizados em ambiente *indoor*, sem luz de fundo. A distância entre os módulos foi variada de acordo com o experimento a ser realizado. Para cada experimento, foi necessário realizar a calibração manual do nível de sinal limite (*threshold*) do receptor, para a correta identificação do início de transmissão da mensagem. Isso, porque a aproximação ou distanciamento dos módulos varia a

intensidade luminosa percebida pelo sensor, o que não é acompanhado pelo valor limite pré-configurado.

A técnica de modulação interna utilizada para o sistema OFDM foi a BPSK, que possui melhor desempenho quanto à recuperação da mensagem recebida.

4.2.1 Distância entre módulos

O primeiro experimento realizado consistiu na variação da distância entre os módulos de transmissão e recepção, para que fosse possível ser feita uma avaliação de como o afastamento entre o emissor e o fotodetector influenciava na recepção e recuperação da mensagem transmitida.

Foram transmitidas sequências de tamanhos 3720 e 3750 bits para os sistemas OOK e OFDM, respectivamente, com período de símbolo de 500 μ s (2000 símbolos por segundo). Para cada distância, foi calculada a média de 10 transmissões. A Figura 30 apresenta os resultados obtidos.

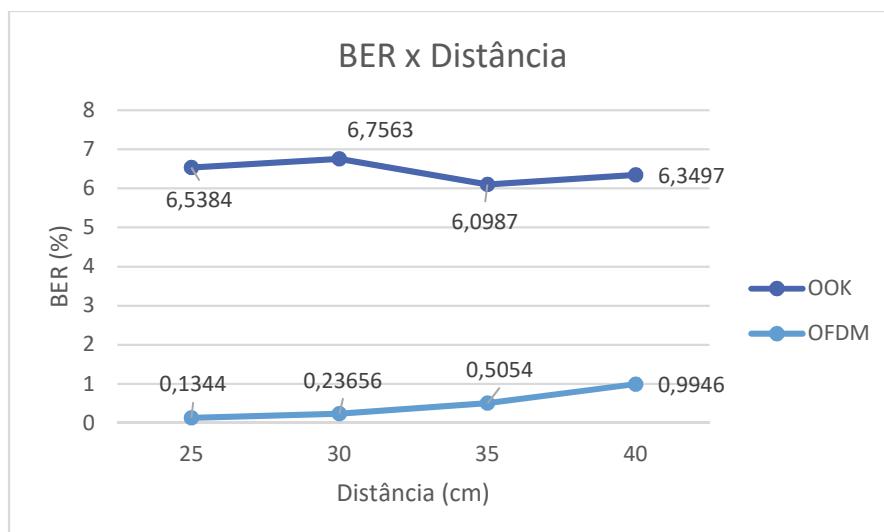


Figura 30 - Gráfico BER x Distância para os sistemas OOK e OFDM.

Nota-se que o sistema OOK apresentou muito mais erros do que o sistema OFDM, que conseguiu recuperar boa parte da mensagem enviada, mantendo o BER abaixo de 1%. Com o aumento da distância entre o LED e o sensor, a porcentagem de erro no sistema OFDM aumentou ligeiramente, enquanto que o sistema OOK manteve-se, em geral, constante.

Esse resultado é compreensível, já que no sistema OOK existem apenas 2 níveis (0 e 1) a serem transmitidos, e o afastamento relativamente pequeno entre os módulos não chega a afetar a decisão entre os símbolos recebidos. Já no sistema OFDM, são transmitidos símbolos que variam entre 256 níveis, o que o torna mais suscetível a erros causados por ruído conforme ocorre o distanciamento. Apesar disso, a recuperação da mensagem original pelo sistema, através das técnicas explicadas na Seção 3.2, garantiu um desempenho melhor do que o sistema OOK, que não dispôs de nenhuma técnica de equalização ou eliminação de interferência.

4.2.2 Período de Símbolo (Taxa de Transmissão)

O segundo experimento teve como objetivo a avaliação do comportamento dos sistemas com relação à taxa de transmissão utilizada. Quanto maior a taxa, menor é o tempo de transmissão de cada símbolo, o que pode resultar mais erros devido ao ruído ou pela própria largura de banda do canal.

Assim como no primeiro experimento, foram transmitidas sequências de 3720 e 3750 bits para os sistemas OOK e OFDM, respectivamente. Para cada taxa de transmissão, foi calculada a média dos resultados de 10 transmissões. A distância entre os módulos transmissor e receptor foi mantida fixa em 25 cm.

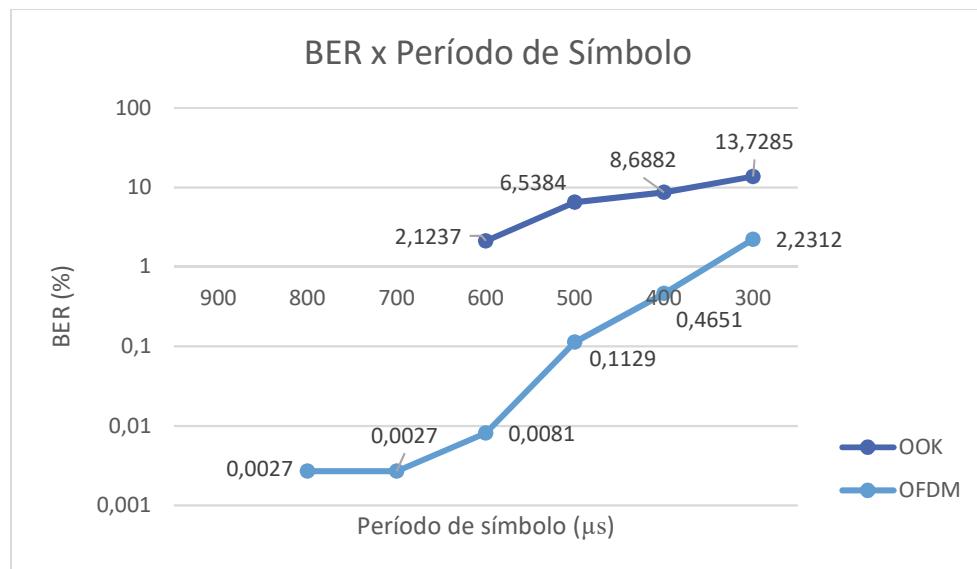


Figura 31 - Gráfico BER x Período de símbolo para os sistemas OOK e OFDM.

Pode-se perceber, pelo gráfico semi-logarítmico da Figura 31, que com o aumento da taxa de transmissão, ambos os sistemas demonstram piora na qualidade da mensagem recebida. Entretanto, o sistema OFDM ainda consegue manter sua taxa de erros bem mais baixa do que a do sistema OOK.

Os valores da curva do sistema OOK para períodos de símbolo menores do que $600\mu s$ não aparecem no gráfico, pois neste ponto não há mais perdas. Já o sistema OFDM ainda possui perdas, mesmo que pequenas, até que o período de símbolo alcance $900\mu s$. Isso mostra que, para este projeto, o uso do sistema OFDM apenas se justificaria para taxas de transmissão com período de símbolo maior do que $700\mu s$.

A maior taxa possível no ATmega328, no entanto, é com período de símbolo por volta de $300\mu s$, já que períodos mais curtos causam instabilidade na leitura devido à frequência de *clock* da placa limitada em 16MHz.

4.2.3 Envio de Imagem

Por último, foi feita a transmissão, através de ambos os sistemas, de uma imagem em escala de cinza de tamanho 240x160 pixels. No MATLAB, foi desenvolvido um código para converter a imagem em uma matriz bidimensional com os valores correspondentes aos tons de cinza, que variam de 0 (preto) a 255 (branco). Em seguida, a matriz foi convertida para uma sequência binária, que foi transmitida pelo sistema óptico, recebida e reconvertida para uma imagem novamente.

Optou-se pela transmissão de uma imagem em escala de cinza devido ao período de símbolo empregado que, no caso do envio de imagem colorida, poderia tornar a transmissão muito demorada - imagens coloridas no MATLAB possuem três vezes mais informação do que imagens em escala de cinza, pois são compostas de uma matriz bidimensional que representa cada uma das três cores: vermelho, verde e azul (RGB).

As Figuras 32 (a) e (b) foram recebidas pelo sistema OOK, com período de símbolo de $500\mu s$ e $600\mu s$, respectivamente. Já as Figuras 32 (b) e (d) foram recebidas pelo sistema OFDM, com os mesmos períodos de símbolo. Foram transmitidos 307200 bits, com a distância fixa de 25 cm entre os módulos transmissor e receptor.

Como esperado, é possível observar que as imagens transmitidas através do sistema OFDM foram muito menos afetadas pelo canal do que as do sistema OOK, apresentando menos pontos com tons de cinza destoantes da imagem.

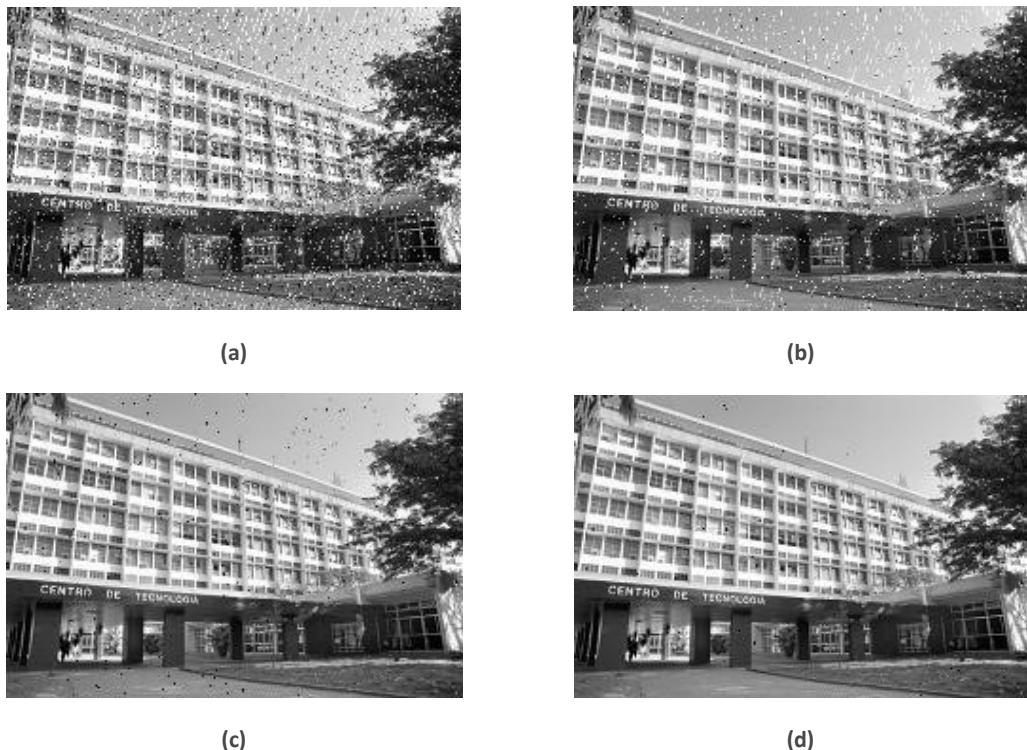


Figura 32 - Imagens enviadas pelos sistemas OOK (a e b) e OFDM (c e d). Para (a) e (c) o período de símbolo utilizado foi 500 μs , e para (b) e (d), 600 μs .

A Tabela 2 evidencia os resultados obtidos nas transmissões. A porcentagem de erros manteve-se aproximadamente igual às obtidas nos experimentos das seções anteriores.

Tabela 2 - Tabela comparativa dos resultados obtidos com os sistemas OOK e OFDM.

Sistema	Período de Símbolo (μs)	Erros	BER (%)
OOK	500	20260	6.5951
	600	6319	2.057
OFDM	500	593	0.19303
	600	70	0.022786

Capítulo 5

Conclusão

5. Conclusão

O protótipo evidenciou, através dos experimentos descritos no Capítulo 4, a vantagem da utilização da técnica de modulação OFDM para as transmissões ópticas. A transmissão do sinal através do uso de subportadoras com inclusão de prefixo cílico, bem como o emprego da técnica de *Zero-Forcing* para equalização do canal, mostrou-se eficaz para a eliminação de interferências, de acordo com o previsto no Capítulo 3.

Foi possível, portanto, transmitir dados de um ponto ao outro, tendo a variação da luz como meio de propagação e o ar como canal, o que valida a aplicação de ondas eletromagnéticas do espectro visível para a comunicação e transferência de dados.

Ainda assim, limitações do projeto, principalmente em relação ao microprocessador utilizado (ATmega328), não permitiram testes com taxas mais altas de transmissão. O tamanho de *buffer* de entrada serial e a memória limitada do microprocessador não possibilitaram a transmissão constante da mensagem, além da frequência de *clock* de 16MHz ter limitado a máxima velocidade de transferência do sistema. Por isso, o uso de um microprocessador superior é recomendado para testes de mais alto desempenho.

Como futura melhoria ao projeto, também é sugerida a aplicação da técnica MMSE (*Minimum Mean Square Error*) para uma melhor equalização do canal, evitando o aumento do nível de ruído branco. Uma forma mais precisa de sincronização é desejável, como a realização da correlação entre as sequências piloto esperada e recebida, de maneira a evitar erros de amostragem. Além disso, recomenda-se a aplicação de uma pré-distorção ao sinal a ser enviado, com a finalidade de compensar a não-linearidade dos componentes semicondutores, como também a inserção de um bloco de *pulse shaping* ao sistema, para um uso mais eficiente da banda.

Como visto no Capítulo 2, as comunicações por luz visível (VLC) são uma forte tendência tecnológica para o descongestionamento do espectro de rádio, que vem se tornando cada vez mais saturado à medida que vem crescendo o número de dispositivos móveis que fazem uso do Wi-Fi e outras tecnologias de comunicação por micro-ondas.

Em especial, o Li-Fi tem se mostrado uma real solução de comunicação por luz visível, com grandes vantagens em relação ao popular Wi-Fi, principalmente no que diz respeito a taxa de transmissão. Contudo, as limitações apresentadas no Capítulo 2,

mostram que o Li-Fi é, não um substituto, mas um complemento às outras tecnologias existentes, que pode vir a contribuir na formação de uma rede heterogênea de grande capacidade, uma promessa das futuras redes 5G.

Referências

- ARDUINO. *PWM*. Disponível em: <https://www.arduino.cc/en/Tutorial/PWM>. Acessado em: 06/02/2017.
- ARMSTRONG, J.; LOWERY, A. J. *Power efficient optical OFDM*, Electronics Letters, Vol. 42 (6), 2006.
- BURNS, R. W. *Communications: An international history of the formative years*. Ed. The Institution of Engineering and Technology, London, United Kingdom, 2004.
- CISCO. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper*. Disponível em: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>. Publicado em: 09/02/2017. Acessado em: 13/02/2017.
- COMPTIA. *Sizing Up the Internet of Things*. Disponível em: <https://www.comptia.org/resources/sizing-up-the-internet-of-things>. Publicado em: 28/08/2015. Acessado em: 9/10/2016.
- DAILYOPTICIAN. *The Electromagnetic Spectrum & UV Rays*. Disponível em: <http://www.dailyoptician.com/andrew-bruce/2015/9/23/the-electromagnetic-spectrum-uv-rays-andrew-bruce>. Publicado em: 23/09/2015. Acessado em: 16/10/2016.
- DIMITROV, S.; HAAS, H. *Principles of LED Light Communications: Towards Networked Li-Fi*. Ed. Cambridge University Press, Cambridge, United Kingdom. 1^a Edição, 2015.
- DINIZ, P.; MARTINS, W.; LIMA M. *Block Transceivers: OFDM and Beyond*. Ed. Morgan & Claypool Publishers, 2012.
- DINIZ, P.; SILVA, E. A. B.; NETTO, S. L. *Processamento Digital de Sinais: Projeto e Análise de Sistemas*. Ed. Bookman, Porto Alegre, Brasil. 2^a Edição, 2014.
- GFELLER, F. R.; BAPST, U. *Wireless In-House Data Communication via Diffuse Infrared Radiation*, Proceedings of the IEEE, Vol. 67 (11), p. 1474-1486, 1979.
- LUCKYLIGHT. *5mm Round Standard T-1 3/4 Type With Flange White LED Technical Data Sheet*. Disponível em: <http://www.soldafria.com.br/datasheet/LL-504-WC2E-W2-3UC.pdf>. Acessado em: 07/08/2016.
- NEZAMALHOSSEINI, S. A; CHEN, L. R.; ZHUGE, Q.; MALEKIHA, M.; MARVASTI, F.; PLANT, D. V. *Theoretical and experimental investigation of direct detection optical OFDM transmission using beat interference cancellation receiver*, Optics Express, Vol. 21 (13), p. 15237-15246, 2013.
- OSA - THE OPTICAL SOCIETY. Disponível em: www.osa.org/en-us/history/gallery/images2/b/bell-photophone-1880. Acessado em: 19/10/2016.
- HAAS, H.; YIN, L.; WANG, Y.; CHEN, C. *What is LiFi?* Journal Of Lightwave Technology, p. 1-12, 2015.

HUTT, D. L.; SNELL, K. J.; BÉLANGER, P. A. *Alexander Graham Bell's Photophone*, Optics & Photonics News, p. 21-25, 1993.

HUURDEMAN, A. A. *The Worldwide History of Telecommunications*. Ed. John Wiley & Sons, Inc., Hoboken, New Jersey, 2003.

JOHNSON, R. C. *Visible light illuminates a new approach for wireless comms*. Disponível em: http://www.eetimes.com/document.asp?doc_id=1256372. Publicado em: 06/07/2010. Acessado em: 01/11/2016.

SAYED, A. H. *Discrete-Time Processing and Filtering*. Ed. Ucla Course Reader Solutions, Los Angeles, p. 1129–1302, 2014.

SCHMIDT, B.; LOWERY, A.; ARMSTRONG, J. *Experimental Demonstrations of Electronic Dispersion Compensation for Long-Haul Transmission Using Direct-Detection Optical OFDM*, Journal of Lightwave Technology, Vol. 26 (1), p. 196-203, 2008.

SHARMA, R.; RAUNAK, K.; SANGANAL, A. *Li-Fi Technology: Transmission of Data Through Light*. International Journal of Computer Technology & Applications, Vol. 5 (1), p. 150-154, 2014.

TEXAS INSTRUMENTS. *OPT101 Monolithic Photodiode and Single-Supply Transimpedance Amplifier*. Disponível em: <http://www.ti.com/lit/ds/symlink/opt101.pdf>. Acessado em: 15/11/2016.

WANG C.; HAIDER F.; GAO X.; YOU X.; YANG Y.; YUAN D.; AGGOUNE H. M.; HASS H.; FLETCHER S.; HEPSAYDIR E. *Cellular Architecture and Key Technologies for 5G Wireless Communication Networks*. IEEE Communications Magazine, p. 122-130, 2014.

WIKIPEDIA. *IEEE 802.11*. Disponível em: https://en.wikipedia.org/wiki/IEEE_802.11. Acessado em: 15/01/2017.

ANEXO A

Código MATLAB – Transmissor OFDM

```
1 -     close all
2 -     clear
3 -     clc
4 -     % create arduino object and connect to board
5 -     arduino = serial('COM3','BaudRate',250000);
6 -     arduino.OutputBufferSize = 64;
7 -
8 -     %
9 -     % Variables
10 -    %
11 -
12 -    M = 64;                      % Input Block Length
13 -    L = 41;                      % Channel Length
14 -    N = M + L - 1;               % Output Block Length
15 -    LENGTH_PILOT = 30;
16 -    LENGTH_FRAMES = 52;
17 -
18 -    %
19 -    % Transmitter
20 -    %
21 -
22 -    load('database.mat','data','data_pilot');    % Load data from database
23 -
24 -    data_input = horzcat(data_pilot,data);        % Join pilot and data sequences
25 -
26 -    % Modulation ---
27 -
28 -    s = 2*data_input - 1;                         % BPSK modulation
29 -
30 -    % Serial-To-Parallel ---
31 -
32 -    for i=0:((M/2-2)-1)
33 -        s_parallel(1+i,:) = downsample(s,(M/2-2),i);
34 -    end
35 -
36 -    [numRow,numCol]=size(s_parallel);
37 -
38 -    delta_s = diag(s_parallel(:,1));
39 -    save('database.mat','delta_s','-append');      % Save pilot sequence
40 -
41 -    % Hermitian symmetry ---
42 -
43 -    % Modify input data disposition to guarantee hermitian symmetry
44 -
45 -    s_hermitian_aux = [zeros(1,numCol);s_parallel;zeros(1,numCol); ...
46 -                       zeros(1,numCol);zeros(1,numCol);flipud(s_parallel)];
47 -    s_hermitian = circshift(s_hermitian_aux,[-M/2 0]);
48 -
49 -    % IDFT ---
50 -
51 -    s_ifft = ifft(s_hermitian);
52 -
53 -    % Add Cyclic Prefix ---
54 -
55 -    s_cp = [eye(M); [eye(L-1) zeros(L-1, M-L+1)]]*s_ifft;
56 -
```

```

57 % Parallel-To-Serial -----
58
59 - x = zeros(1, (M+L-1)*numCol);
60
61 - for i=0:(M+L-2)
62 -     shift_s = zeros(1, (M+L-1)*numCol);
63 -     up_s = zeros(1, (M+L-1)*numCol);
64 -     up_s = upsample(s_cp(1+i,:), M+L-1);
65 -     shift_s(1+i:end) = up_s(1:end-i);
66 -     x = x + shift_s;
67 - end
68
69 % Send to Arduino -----
70
71 % Data is sent to Arduino through serial port as ASCII characters
72 % Data values must be scaled to fit the range of 256 (0 to 255)
73
74 - message_aux = x + abs(min(x));
75 - message_aux = uint8(255*message_aux/(max(message_aux)));
76 - [numRowMessage, numColMessage] = size(message_aux);
77
78 - save('database.mat','numRowMessage','numColMessage','-append');
79
80 - message = zeros(numRowMessage, numColMessage*2);
81
82 % Arduino ASCII only accepts range of 128, so the 256 are split in two
83
84 - for i=0:numColMessage-1
85 -     if message_aux(1,1+i)>127
86 -         message(1,1+2*i) = 43;
87 -         message(1,2+2*i) = 255-message_aux(1,1+i);
88 -     else
89 -         message(1,1+2*i) = 45;
90 -         message(1,2+2*i) = message_aux(1,1+i);
91 -     end
92 - end
93
94 - message_char = char(message);
95 - NUMBER_FRAMES = ceil(numColMessage/(LENGTH_FRAMES/2));
96
97 % The message is ready to be sent to Arduino
98
99 - fopen(arduino);
100
101 - while(1)
102
103 -     command = input('[Send=1] [Exit=2] ');
104
105 -     if command==1
106 -         for i=1:NUMBER_FRAMES
107 -             sent=0;
108 -             fprintf(arduino, '%s', ...
109 -                     message_char(1,1+(LENGTH_FRAMES*(i-1)):LENGTH_FRAMES*i));
110 -             while(sent~=1)
111 -                 sent = fscanf(arduino, '%i');
112 -             end
113 -         end
114 -     end
115
116 -     if command==2
117 -         break;
118 -     end
119
120 - end
121
122 - fclose(arduino);
123

```

Código MATLAB – Receptor OFDM

```
1 -     close all
2 -     clear
3 -     clc
4 -     % create arduino object and connect to board
5 -     arduino = serial('COM4','BaudRate',250000);
6 -     set(arduino, 'TimeOut', 20);
7 -
8 -     %
9 -     % Variables
10 -    %
11
12 -    M = 64;                      % Input Block Length
13 -    L = 41;                      % Channel Length
14 -    N = M + L - 1;               % Output Block Length
15 -    LENGTH_PILOT = 30;
16 -    LENGTH_FRAMES = 52;
17 -
18 -    %
19 -    % Receiver
20 -    %
21
22 -    load('database.mat');        % Load data and image information from database
23 -    i=0;
24 -    command=1;
25
26 -    % Receive message from Arduino
27
28 -    fopen(arduino);
29
30 -    while(1)
31
32 -        if command==1
33 -            while(i<numColMessage)
34 -                message(1,1+i) = fscanf(arduino,'%i');
35 -                disp(1+i);
36 -                i = i + 1;
37 -            end
38 -        elseif command==2
39 -            break;
40 -        end
41
42 -        if i==numColMessage
43 -            command = input('[Resume=1] [Exit=2] ');
44 -            i=0;
45 -        end
46
47 -    end
48
49 -    fclose(arduino);
50
51 -    y = message - abs(min(message));           % Scale data to fit the range
52 -    y = uint8(255*y/(max(y)));                 % of 0 to 255 values
53
54 -    % Serial-To-Parallel -----
55
56 -    for i=0:(N-1)
57 -        y_parallel(1+i,:) = downsample(y,N,i);
58 -    end
```

```

60 % Remove Cyclic Prefix -----
61
62 - y_n_cp = y_parallel(1:M,:);
63
64 % DFT -----
65
66 - y_fft = fft(y_n_cp);
67
68 % Remove Hermitian Symmetry -----
69
70 - y_hermitian_aux = circshift(y_fft,[M/2 0]);
71 - y_n_hermitian = y_hermitian_aux(2:(M/2-1),:);
72
73 % Equalization -----
74
75 % It was necessary to shift values vertically, according to the pilot
76 % values in order to do the right channel equalization
77
78 - [a,b] = size(y_n_hermitian);
79 - pilot_norm = y_n_hermitian(:,1)/norm(y_n_hermitian(:,1));
80 - variation = mean(diag(delta_s))-mean(pilot_norm);
81
82 - for i=1:b
83 -     y_norm(:,i) = y_n_hermitian(:,i)/norm(y_n_hermitian(:,i));
84 -     y_received(:,i) = y_norm(:,i)-abs(variation);
85 - end
86
87 - lambda = real((delta_s^-1)*y_received(:,1));
88 - delta = diag(lambda);
89 - s_eq = (delta^-1)*y_received(:,2:end);    % Zero-Forcing Equalization
90
91 % Detection & Parallel-To-Serial -----
92
93 - s_re = real(s_eq);
94
95 - s_detection = zeros(size(data));
96
97 - s_detection(s_re < 0) = -1;                % Decision between BPSK symbols
98 - s_detection(s_re >= 0) = 1;
99
100 % Demodulation -----
101
102 - data_received = (s_detection + 1)/2; % BPSK Demodulation
103
104 % Result -----
105
106 - [ne,nBER] = biterr(data, data_received);
107 - result = ['Errors=', num2str(ne),'; BER=', num2str(nBER*100), '%'];
108 - disp(result);
109
110 % Image Received -----
111
112 % Show received image
113
114 - for n=1:numRowImage
115 -     I(n,1:8) = data_received(1,1+8*(n-1):8*n);
116 - end
117
118 - for k=1:numRowImage
119 -     aux(k,:) = strcat(int2str(I(k,:)));
120 -     aux2(k,:) = uint8(bin2dec(aux(k,:)));
121 - end
122
123 - imageReceived = reshape(aux2, sizeImage);
124
125 - imwrite(imageReceived, 'ct_500_OFDM.jpg');

```

ANEXO B

Código Arduino – Transmissor OFDM

```
#include <TimerOne.h>
#include <stdlib.h>

#define LENGTH_PACKETS 104

int ledPin = 9;           // LED connected to digital pin 9
int delayTime = 500;      // Period of Symbol (Transmission Rate)
char a;
int message[LENGTH_PACKETS];
char info[LENGTH_PACKETS];
int indice = 0;
int i = 0;
int counter = 0;

void setup()
{
    Serial.begin (250000);
    pinMode(ledPin, OUTPUT);           // sets the pin as output
    TCCR1B = TCCR1B & B11111000 | B00000001; // set timer 1 divisor to 1 for
                                            // PWM frequency of 31372.55 Hz
}

void loop()
{
    if (Serial.available() > 0) // Receive data from computer
    {
        a = Serial.read();
        delay(1);
        info[indice] = a;
        indice++;

        if ((Serial.available() <= 0)&&(counter<LENGTH_PACKETS)) // Received
        {
            // Identify ASCII characters received
            for (int i=0; i<indice; i++)
            {
                if ((i%2)==0)
                {
                    if (info[i]=='+') message[counter] = 255 -((int)info[i+1]);
                    else if (info[i]=='-') message[counter] = (int) info[i+1];
                    counter=counter+1;
                }
            }
            memset(&info,0,sizeof(info));
            indice=0;
            Serial.println('1'); // Send confirmation of received data to computer
        }
    }
    if (counter==LENGTH_PACKETS)
    {
        counter=0;
        analogWrite(ledPin,0);           // Transmit starting bit of message
        delayMicroseconds(delayTime/4);

        while(counter<LENGTH_PACKETS)    // Transmit OFDM symbols
        {
            analogWrite(ledPin, message[counter]);
            delayMicroseconds(delayTime);
            counter=counter+1;
        }
        if (counter==LENGTH_PACKETS)
        {
            counter=0;
            indice=0;
            memset(&message,0,sizeof(message));
        }
    }
    else if (counter<LENGTH_PACKETS) analogWrite(ledPin, 255); // Transmit HIGH level
}
}
```

Código Arduino – Receptor OFDM

```
#include <TimerOne.h>

#define LENGTH_PACKETS 104

int threshold = 802;          // Threshold
int period = 510;             // Message Sample rate (microseconds)
int period_default = 250;     // Default Sample rate before message (microseconds)
int value=1023;
int i=0;
int message[LENGTH_PACKETS];
boolean flag = false;
boolean valuePrevious = false;
boolean valueReceived = false;

void setup()
{
    Serial.begin (250000);
    Timer1.initialize(period_default);
    Timer1.attachInterrupt(SampleInput);
}

void SampleInput()
{
    value = analogRead(A1);    // Read value from sensor OPT101

    if (value>=threshold) valueReceived = false;
    else valueReceived = true;

    if (flag) // Save value if message are being received
    {
        message[i] = value;
        i=i+1;
        if (i==LENGTH_PACKETS)
        {
            i=0;
            flag = false;
            valuePrevious = valueReceived;

            // Send values to computer
            for (int counter=0; counter<LENGTH_PACKETS; counter++) Serial.println(message[counter]);

            memset(&message,0,sizeof(message));
            Timer1.setPeriod(period_default);
        }
    }

    if ((valueReceived)&&(!valuePrevious)&&(!flag)) // If begin of message
    {
        Timer1.setPeriod(period);    // Set sample rate of message
        flag = true;                // Next loop, begin sampling message
    }

    valuePrevious = valueReceived;
}

void loop(){}
```

ANEXO C

Código MATLAB – Simulação OFDM

```
1 -      close all
2 -      clear
3 -      clc
4 -
5 -      %-----
6 -      % Variables
7 -      %-----
8 -
9 -      M = 64;                      % Input Block Length
10 -     L = 41;                      % Channel Length
11 -     N = M + L - 1;               % Output Block Length
12 -     vSNR = -20:0.5:20;           % SNR range values (dB)
13 -     n = 5000;                    % Number of channels
14 -     nBER = 0;
15 -
16 -     totalBER_4qam = zeros(1,length(vSNR));
17 -     totalBER_bpsk = zeros(1,length(vSNR));
18 -
19 -     for mOrder = [2,4]
20 -         if mOrder==2
21 -             numbBits = 2400;
22 -             numbBits_pilot = 30;
23 -         else
24 -             clearvars -except mOrder totalBER_bpsk totalBER_4qam M L N vSNR n
25 -             numbBits = 4800;
26 -             numbBits_pilot = 60;
27 -         end
28 -
29 -         numbBits_size = 2400;
30 -
31 -         %-----
32 -         % Transmitter
33 -         %-----
34 -
35 -
36 -         data_pilot = randi([0 1],1,numbBits_pilot); % Generate random pilot and
37 -         data = randi([0 1],1,numbBits);              % random data
38 -
39 -         data_input = horzcat(data_pilot,data);       % Join pilot and data sequences
40 -
41 -         % Modulation -----
42 -
43 -         % 4-QAM modulation
44 -         if (mOrder==4)
45 -             data_aux = bi2de(vec2mat(data_input,2),2,'left-msb');
46 -             s = (sqrt(2)/2)*qammod(data_aux,mOrder,0);
47 -         % BPSK modulation
48 -         else
49 -             s = 2*data_input - 1;
50 -         end
51 -
52 -
53 -         % Serial-To-Parallel -----
54 -
55 -         for i=0:(M/2-2)-1
56 -             s_parallel(1+i,:) = downsample(s,(M/2-2),i);
57 -         end
58 -
59 -         [numRow,numCol]=size(s_parallel);
60 -
61 -         delta_s = diag(s_parallel(:,1));
62 -         save('database.mat','delta_s','-append');    % Save pilot sequence
```

```

64 % Hermitian symmetry -----
65
66 % Modify input data disposition to guarantee hermitian symmetry
67
68 - if mOrder==4
69 -     s_hermitian_aux = [zeros(1,numCol);s_parallel;zeros(1,numCol); ...
70 -                         zeros(1,numCol);zeros(1,numCol); ...
71 -                         conj(flipud(s_parallel))];
72 -     s_hermitian = circshift(s_hermitian_aux,[-M/2 0]);
73 - else
74 -     s_hermitian_aux = [zeros(1,numCol);s_parallel;zeros(1,numCol); ...
75 -                         zeros(1,numCol);zeros(1,numCol); ...
76 -                         flipud(s_parallel)];
77 -     s_hermitian = circshift(s_hermitian_aux,[-M/2 0]);
78 - end
79
80 % IDFT -----
81
82 - s_ifft = ifft(s_hermitian);
83
84 % Add Cyclic Prefix -----
85
86 - s_cp = [eye(M); [eye(L-1) zeros(L-1, M-L+1)]]*s_ifft;
87
88 %-----
89 % Receiver
90 %-----
91
92 - for noise = 1:length(vSNR)
93 -     SNR=vSNR(noise);
94 -     disp(SNR);
95 -     for channel = 1:n
96
97 -         h = sqrt(1/2)*(randn(1,L));      % Generate random channel
98 -         H = zeros(M,N);
99
100 -         % Build Toeplitz matrix
101 -         for i = 1:M
102 -             H(i,i:L+i-1) = h;
103 -         end
104
105 -         % Transmit through channel and add white gaussian noise
106 -         for block=1:numCol
107 -             z(:,block) = H*s_cp(:,block);
108 -             y(:,block) = awgn(z(:,block),SNR,'measured');
109 -         end
110
111         % DFT -----
112
113 -         y_fft = fft(y);
114
115         % Remove Hermitian Symmetry -----
116
117 -         y_hermitian_aux = circshift(y_fft,[M/2 0]);
118 -         y_n_hermitian = y_hermitian_aux(2:(M/2-1),:);
119
120         % Equalization -----
121
122         % Zero-Forcing Equalization
123
124 -         lambda = (delta_s^-1)*y_n_hermitian(:,1);
125 -         delta = diag(lambda);
126 -         s_eq = (delta^-1)*y_n_hermitian(:,2:end);
127
128         % Detection & Parallel-To-Serial -----
129
130 -         s_re = real(s_eq);
131 -         s_im = imag(s_eq);
132
133 -         s_detection = zeros(1,numBIts_size);
134

```

```

135 % Decision between 4-QAM symbols
136 if mOrder==4
137     s_detection(s_re < 0 & s_im < 0) = -0.7071 - 0.7071i;
138     s_detection(s_re >= 0 & s_im >= 0) = 0.7071 + 0.7071i;
139     s_detection(s_re < 0 & s_im >= 0) = -0.7071 + 0.7071i;
140     s_detection(s_re >= 0 & s_im < 0) = 0.7071 - 0.7071i;
141
142 % Decision between BPSK symbols
143 else
144     s_detection(s_re < 0) = -1;
145     s_detection(s_re >= 0) = 1;
146 end
147
148 % Demodulation -----
149
150 % 4-QAM Demodulation
151 if (mOrder==4)
152     s_demod = qamdemod(s_detection,mOrder);
153     s_aux = de2bi(s_demod,2,'left-msb');
154
155 data_received = zeros(1,2*length(s_aux));
156
157 for i=0:length(s_aux)-1
158     for j=1:2
159         data_received(1,2*i+j) = s_aux(i+1,j);
160     end
161 end
162
163 [ne,nBER] = biterr(data_received, data);
164 totalBER_4qam(1,noise) = totalBER_4qam(1,noise) + nBER;
165
166 % BPSK Demodulation
167 else
168     data_received = (s_detection + 1)/2;
169     [ne,nBER] = biterr(data, data_received);
170     totalBER_bpsk(1,noise) = totalBER_bpsk(1,noise) + nBER;
171 end
172
173 end % Canal Loop
174
175 end % SNR Loop
176
177 end % mOrder Loop
178
179 totalBER_4qam = totalBER_4qam/n; % Average BER 4-QAM
180 totalBER_bpsk = totalBER_bpsk/n; % Average BER BPSK
181
182 %-----
183 % Analysis
184 %
185
186 % Bit Error Rate -----
187 semilogy(vSNR,totalBER_4qam, '-b',vSNR,totalBER_bpsk, '-r')
188 legend('4-QAM','BPSK')
189 title('OFDM Overlap-and-Save')
190 xlabel('SNR (dB)')
191 ylabel('BER')
192 axis([-20 20 0.001 1])
193 grid on

```

ANEXO D

Código MATLAB – Transmissor OOK

```
1 - close all
2 - clear
3 - clc
4 - % create arduino object and connect to board
5 - arduino = serial('COM3','BaudRate',250000);
6 - arduino.OutputBufferSize = 64;
7 -
8 %-----
9 % Variables
10 %
11 LENGTH_FRAMES = 30;
12 %
13 %
14 %
15 % Transmitter
16 %
17
18 load('database.mat','data');
19
20 [~, numCol] = size(data);
21
22 NUMBER_FRAMES_ARDUINO = ceil(numCol/(2*LENGTH_FRAMES));
23 FILL_SYMBOLS = (NUMBER_FRAMES_ARDUINO*2*LENGTH_FRAMES)-(numCol);
24 data_input = horzcat(data,ones(1,FILL_SYMBOLS));
25
26 [numRow, numCol] = size(data_input);
27 NUMBER_FRAMES = numCol/LENGTH_FRAMES;
28 message = int2str(data_input.');
29
30 fopen(arduino);
31
32 while(1)
33
34     command = input('[Send=1] [Exit=2] ');
35
36     if command==1
37         for i=1:NUMBER_FRAMES
38             sent=0;
39             fprintf(arduino,'%s', ...
40                     message(1+(LENGTH_FRAMES*(i-1)):LENGTH_FRAMES*i,1));
41             while(sent~=1)
42                 sent = fscanf(arduino,'%i');
43             end
44         end
45     end
46
47     if command==2
48         break;
49     end
50
51 end
52
53 fclose(arduino);
54
```

Código MATLAB – Receptor OOK

```
1 -     close all
2 -     clear
3 -     clc
4 -     % create arduino object and connect to board
5 -     arduino = serial('COM4','BaudRate',250000);
6 -     set(arduino, 'TimeOut', 20);
7 -     arduino.InputBufferSize = 52;
8 -
9 -     %-----
10 -    % Variables
11 -    %-----
12 -
13 -    LENGTH_FRAMES = 30;
14 -
15 -    %-----
16 -    % Receiver
17 -    %-----
18 -
19 -    load('database.mat');
20 -    [numRow, numCol] = size(data);
21 -
22 -    NUMBER_FRAMES_ARDUINO = ceil(numCol/(2*LENGTH_FRAMES));
23 -    FILL_SYMBOLS = (NUMBER_FRAMES_ARDUINO*2*LENGTH_FRAMES)-(numCol);
24 -
25 -    NUMBER_BITS = det(diag(size(data)))+FILL_SYMBOLS;
26 -
27 -    i=0;
28 -    command=1;
29 -
30 -    fopen(arduino);
31 -
32 -    while(1)
33 -
34 -        if command==1
35 -            while(i<NUMBER_BITS)
36 -                message(1,1+i) = fscanf(arduino, '%i');
37 -                i = i + 1;
38 -            end
39 -        elseif command==2
40 -            break;
41 -        end
42 -
43 -        if i==NUMBER_BITS
44 -            command = input('[Resume=1] [Exit=2] ');
45 -            i=0;
46 -        end
47 -
48 -    end
49 -
50 -    fclose(arduino);
51 -
52 -    data_received = uint8(message);
53 -
54 -    % Result -----
55 -
56 -    [ne,nBER] = biterr(data,data_received);
57 -    result = ['Errors=', num2str(ne),'; BER=', num2str(nBER*100),'%'];
58 -    disp(result);
59 -
60 -    % Image Received -----
61 -
62 -    for n=1:numRowImage
63 -        I(n,1:8) = data_received(1,1+8*(n-1):8*n);
64 -    end
65 -
66 -    for k=1:numRowImage
67 -        aux(k,:) = strcat(int2str(I(k,:)));
68 -        aux2(k,:) = uint8(bin2dec(aux(k,:)));
69 -    end
70 -
71 -    imageReceived = reshape(aux2, sizeImage);
72 -
73 -    imwrite(imageReceived,'ct_500.jpg');
74 -
```

ANEXO E

Código Arduino – Transmissor OOK

```
#include <TimerOne.h>
#include <stdlib.h>

#define LENGTH_FRAMES 60

int ledPin = 9;           // LED connected to digital pin 9
int delayTime = 500;      // Period of Symbol (Transmission Rate)
char a;
int message[LENGTH_FRAMES];
char info[LENGTH_FRAMES];
int indice = 0;
int i = 0;
int counter = 0;

void setup()
{
    Serial.begin (250000);
    pinMode(ledPin, OUTPUT);          // sets the pin as output
    TCCR1B = TCCR1B & B11111000 | B00000001; // set timer 1 divisor to 1 for
}                                         // PWM frequency of 31372.55 Hz

void loop()
{
    if (Serial.available() > 0) // Receive data from computer
    {
        a = Serial.read();
        delay(1);
        info[indice] = a;
        indice++;

        if ((Serial.available() <= 0)&&(counter<LENGTH_FRAMES)) // Received
        {
            for (int i=0; i<indice; i++)
            {
                message[counter] = info[i]-'0';
                counter=counter+1;
            }
            memset(&info,0,sizeof(info));
            indice=0;
            Serial.println('1'); // Send confirmation of received data to computer
        }
    }
    if (counter==LENGTH_FRAMES)
    {
        counter=0;
        analogWrite(ledPin,0);          // Transmit starting bit of message
        delayMicroseconds(delayTime/4);

        while(counter<LENGTH_FRAMES) // Transmit OFDM symbols
        {
            analogWrite(ledPin, 255*((int) message[counter]));
            delayMicroseconds(delayTime);
            counter=counter+1;
        }
        if (counter==LENGTH_FRAMES)
        {
            counter=0;
            indice=0;
            memset(&message,0,sizeof(message));
        }
    }
    else if (counter<LENGTH_FRAMES) analogWrite(ledPin, 255); // Transmit HIGH level
}
```

Código Arduino – Receptor OOK

```
#include <TimerOne.h>

#define LENGTH_FRAMES 60

int threshold = 801;           // Threshold
int period = 510;             // Message Sample rate (microseconds)
int period_default = 250;      // Default Sample rate before message (microseconds)
int value=1023;
int i=0;
char message[LENGTH_FRAMES];
boolean flag = false;
char valuePrevious = '0';
char valueReceived = '0';

void setup()
{
    Serial.begin (250000);
    Timer1.initialize(period_default);
    Timer1.attachInterrupt(SampleInput);
}

void SampleInput()
{
    value = analogRead(A1);    // Read value from sensor OPT101

    if (value>=threshold) valueReceived = '1';
    else valueReceived = '0';

    if (flag) // Save value if message are being received
    {
        message[i] = valueReceived;
        i=i+1;
        if (i==LENGTH_FRAMES)
        {
            i=0;
            // Send values to computer
            for (int counter=0; counter<LENGTH_FRAMES; counter++) Serial.println(message[counter]);
            memset(message,0,sizeof(message));
            flag = false;
            valuePrevious = valueReceived;
            threshold=threshold+95;          // Set threshold to default value
            Timer1.setPeriod(period_default); // Set default sample rate
        }
    }

    if ((valueReceived=='0')&&(valuePrevious=='1')&&(!flag)) // If begin of message
    {
        Timer1.setPeriod(period); // Set sample rate of message
        threshold=threshold-95;  // Set threshold to sampling value (Decision threshold)
        flag = true;            // Next loop, begin sampling message
    }

    valuePrevious = valueReceived;
}

void loop(){
}
```