

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
PROJETO DE GRADUAÇÃO**

**EDUARDO RIBEIRO VARGAS DUARTE**

**DESENVOLVIMENTO DE UM PROTÓTIPO DE  
COMUNICAÇÃO POR LUZ VISÍVEL UTILIZANDO  
MODULAÇÃO DE AMPLITUDE**

VITÓRIA – ES  
SETEMBRO/2016

EDUARDO RIBEIRO VARGAS DUARTE

**DESENVOLVIMENTO DE UM PROTÓTIPO DE COMUNICAÇÃO  
POR LUZ VISÍVEL UTILIZANDO MODULAÇÃO DE AMPLITUDE**

Parte manuscrita do Projeto de Graduação do aluno **Eduardo Ribeiro Vargas Duarte**, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Dr. Jair Adriano Lima Silva

VITÓRIA – ES  
SETEMBRO/2016

EDUARDO RIBEIRO VARGAS DUARTE

**DESENVOLVIMENTO DE UM PROTÓTIPO DE COMUNICAÇÃO  
POR LUZ VISÍVEL UTILIZANDO MODULAÇÃO DE AMPLITUDE**

Parte manuscrita do Projeto de Graduação do aluno **Eduardo Ribeiro Vargas Duarte**, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovada em 29 de Setembro de 2016.

**COMISSÃO EXAMINADORA:**

---

**Prof. Dr. Jair Adriano Lima Silva**  
Universidade Federal do Espírito Santo  
Orientador

---

**Prof Dr. André Ferreira**  
Universidade Federal do Espírito Santo  
Examinador

---

**Profa. Dra. Maria José Pontes**  
Universidade Federal do Espírito Santo  
Examinadora

Dedico esse trabalho a minha família pelo apoio durante todos esses anos de faculdade. Aos amigos que, de alguma forma, contribuíram para este trabalho. Aos professores pelas lições ensinadas dentro e fora da sala de aula. Ao orientador pelo auxílio nas etapas deste trabalho.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus pela oportunidade de realizar este trabalho. A minha família pelo incentivo, principalmente nas horas mais difíceis. Ao meu orientador que sempre estava disposto a ajudar e tirar dúvidas relativas ao trabalho. Aos amigos que estiveram presentes e que ajudaram que essa difícil caminhada fosse um pouco mais agradável. Enfim, a todos que de alguma forma contribuíram que este trabalho pudesse ser finalizado.

## **RESUMO**

Esta monografia apresenta um projeto de um protótipo de Comunicação por Luz Visível (VLC) que utiliza diodos emissores de luz (LEDs) para comunicação e iluminação. Nesse tipo de sistema, os LEDs funcionam em uma frequência tão elevada que o olho humano não é capaz de notar mudanças em seu comportamento. Porém um fotodiodo é capaz de detectar essas variações e, com o auxílio de um sistema embarcado, é possível utilizar esse conjunto para a transmissão de dados. Na primeira parte desse projeto, serão apresentadas as motivações que levaram a escolha deste tema, mostrando em que aspectos VLC pode ser útil no cenário das Telecomunicações atual. Em seguida serão detalhados os procedimentos necessários para a montagem do protótipo, a metodologia que será utilizada e os resultados obtidos. Por fim, será descrito as possibilidades de trabalhos futuros e de possíveis melhorias para o melhor desempenho do protótipo.

## LISTA DE FIGURAS

Figura 1 – Espectro de luz visível.....	12
Figura 2 – Faixas de frequência definidas pela ANATEL.....	14
Figura 3 – Representação do funcionamento do fotofone.....	15
Figura 4 – Comunicação do Trânsito.....	18
Figura 5 - VLC em Hospitais.....	19
Figura 6 – Diagrama de blocos do sistema VLC em aviões .....	20
Figura 7 – LED de cor branca utilizado no projeto.....	22
Figura 8 – Ângulo de visão de um LED.....	23
Figura 9 – Fotodiodos utilizados no projeto .....	25
Figura 10 – Representação de diodos do tipo P-N e PIN.....	26
Figura 11 – Launchpad MSP-EXP430G2.....	27
Figura 12 – Diagrama de blocos de um microcontrolador.....	28
Figura 13 – Interface do Code Composer Studio.....	29
Figura 14 – Interface do MATLAB.....	30
Figura 15 – Circuito do receptor roteado pelo EAGLE.....	31
Figura 16 – Diagrama funcional do modelo de VLC.....	32
Figura 17 – Mensagem capturada no transmissor.....	34
Figura 18 – Circuito eletrônico utilizado no projeto.....	35
Figura 19 – Circuito do emissor e receptor no EAGLE.....	36
Figura 20 – Arquivo texto os valores gravados do CCS.....	37
Figura 21 – Sinal distorcido no receptor.....	39
Figura 22 – Sinal no receptor para uma distância de 90 cm.....	40
Figura 23 – Sinal no receptor para um ambiente com alto índice de iluminação.....	41
Figura 24 – Sinal no receptor para um ambiente com as condições favoráveis .....	42
Figura 25 – Janela de comando do MATLAB com a mensagem enviada.....	44
Figura 26 – Janela de comando do MATLAB com a mensagem enviado com erro.....	45

## **LISTA DE GRÁFICOS**

Gráfico 1 – Relação entre comprimento de onda e sensibilidade do olho humano.....24



## **LISTA DE TABELAS**

Tabela 1 - Tabela ASCII.....	43
------------------------------	----

## **LISTA DE ABREVIATURAS E SIGLAS**

ADC	Conversor Analógico-Digital
ANATEL	Agência Nacional de Telecomunicações
ASK	Modulação de amplitude
BER	Taxa de erro de bit
CICTR	Center for Information & Communications Technology Research
CCS	Code Composer Studio
DTC	Data Transfer Control
EAGLE	Easy Applicable Graphical Layout Editor
IDE	Integrated Development Environment
LED	Diodo Emissor de Luz
MATLAB	Matrix Laboratory
MIMO	Multiple Input Multiple Output
OOK	On-Off Keying
PCB	Placa de Circuito Impresso
PLC	Power Line Communications
PWM	Modulação por Largura de Pulso
RAM	Memória de Acesso Aleatório
RF	Radiofrequência
SISO	Single Input Single Output
UFES	Universidade Federal do Espírito Santo
VLC	Comunicação por Luz Visível
VLCC	Consórcio de Comunicação por Luz Visível

## LISTA DE SÍMBOLOS

$I_v$	Intensidade luminosa (cd)
$\Phi_v$	Fluxo luminoso (lm)
$\lambda_{max}$	Comprimento de onda máximo (nm)
T	Temperatura de cor (K)
F	Frequência (Hz)

# SUMÁRIO

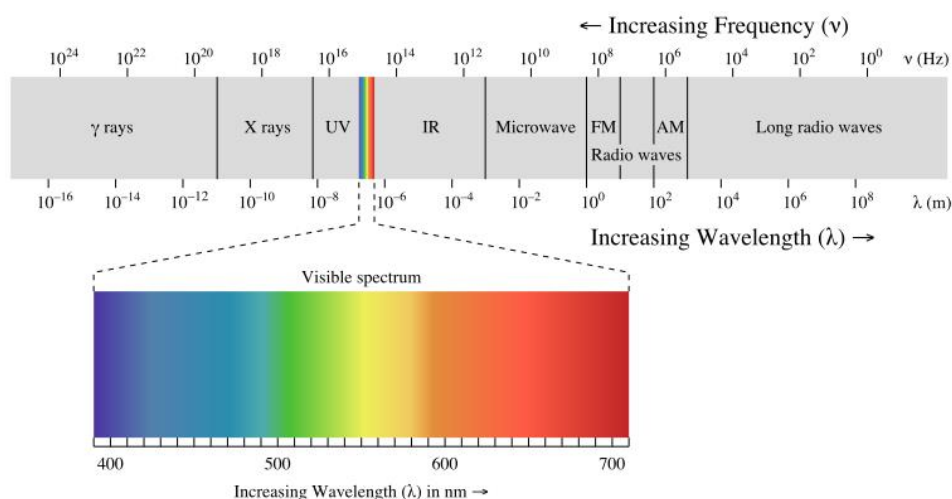
<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>12</b>
1.1	Justificativa.....	13
1.2	Possíveis Aplicações de VLC.....	17
1.2.1	Troca de informações no trânsito de veículos.....	17
1.2.2	Uso em ambientes hospitalares.....	18
1.2.3	Acesso à internet em aeronaves.....	20
<b>2</b>	<b>OBJETIVOS GERAIS E ESPECÍFICOS .....</b>	<b>21</b>
<b>3</b>	<b>DESCRIÇÃO DO SISTEMA .....</b>	<b>22</b>
3.1	O Dispositivo de Conversão Eletro-Óptico.....	22
3.2	O Dispositivo de Conversão Óptico-Elétrico.....	24
3.3	O Microcontrolador MSP-EXP430G2 Usado como Conversor ADC.....	26
3.4	O Ambiente de Desenvolvimento para Microcontroladores.....	28
3.5	Leitura de Variáveis via Linguagem de Computação MATLAB.....	29
3.6	O Software usado no Projeto das Placas de Circuito Impresso .....	30
<b>4</b>	<b>METODOLOGIA E ETAPAS DE DESENVOLVIMENTO DO PROJETO .....</b>	<b>32</b>
4.1	Descrição da Transmissão de Sinal VLC.....	32
4.2	Circuito Eletrônico Implementado.....	34
4.3	Descrição da Recepção de Sinal VLC.....	37
<b>5</b>	<b>ANÁLISE DOS RESULTADOS DOS EXPERIMENTOS .....</b>	<b>39</b>
5.1	Avaliação do Efeito do Atraso na Transmissão.....	39
5.2	Avaliação do Efeito da Distância de Comunicação.....	40
5.3	Avaliação do Efeito do Ruído do Ambiente.....	40
5.4	Comunicação VLC em Condições Favoráveis.....	41
5.5	Análise das Limitações do Microcontrolador e da Coleta dos Dados .....	42
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>46</b>
<b>7</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>48</b>

# 1 INTRODUÇÃO

Este projeto apresenta o desenvolvimento de um protótipo de Comunicação por Luz Visível (VLC) utilizando o Launchpad MSP-EXP430G2, produzido pela Texas Instruments, e as etapas necessárias para sua implementação e funcionamento.

VLC é uma tecnologia de comunicação que utiliza do espectro de luz visível entre 400 e 700 nm para transmitir informação e para iluminação. Essa tecnologia representa um ramo da comunicação ótica sem fio. A Figura 1 ilustra a localização do espectro de luz visível.

Figura 1: Espectro de luz visível



Fonte: RONAN, 2007

A comunicação sem fio em ambientes fechados ganhou atenção mundial nos últimos anos, especialmente pelo fato de fornecer transmissão em alta velocidade para aplicações em tempo real. Dentre as próximas gerações de tecnologia sem fio, a comunicação por luz visível (VLC) é apontada como alternativa para serviços como televisão 3D (3DTV) e televisão de alta definição (UHDTV). VLC pode oferecer simultaneamente transmissão de dados e iluminação sem interferência eletromagnética (1).

A comunicação por luz visível utiliza lâmpadas fluorescentes ou LEDs para a transmissão de dados. Fotodiodos geralmente são utilizados como receptores do sistema, capturando as informações enviadas pelo transmissor.

Microcontroladores foram utilizados no emissor e no receptor para fazer a conversão da mensagem em níveis de tensão (representado por zeros e uns) na transmissão e para converter esses níveis de tensão novamente na mensagem original na recepção.

Alguns outros componentes eletrônicos como resistores, amplificadores operacionais e baterias também foram utilizados neste projeto para conseguir que a mensagem original fosse recuperada com sucesso na recepção.

## **1.1 Justificativa**

Com o considerável aumento da demanda de transmissão de dados ao longo dos últimos anos, novas tecnologias de informação começaram a ser estudadas para suprir a necessidade dos sistemas de comunicação sem fio. Essa expansão dos sistemas de comunicação esbarra no espectro de alocação de frequências, que possui um número limitado de faixas para que cada sistema funcione.

No Brasil, a ANATEL é o órgão responsável pela administração e pela manutenção do plano de atribuição, destinação e distribuição de radiofrequências (2). Pela Figura 2 é possível ver a atribuição de frequências no Brasil para diferentes sistemas de comunicação. Na figura é possível perceber que o espectro de frequências já está bem saturado, sendo que a grande maioria das faixas já possui uma atribuição para algum tipo de sistema de Telecomunicação.

A utilização de uma mesma faixa de frequência para diferentes tipos de sinais pode causar problemas de interferência, o que faz com que a informação seja transmitida de forma errada ou incompleta. Desse modo, deve-se evitar a utilização simultânea da mesma faixa para diferentes sinais sempre que possível, para que não haja possibilidade de ocorrer tais problemas.



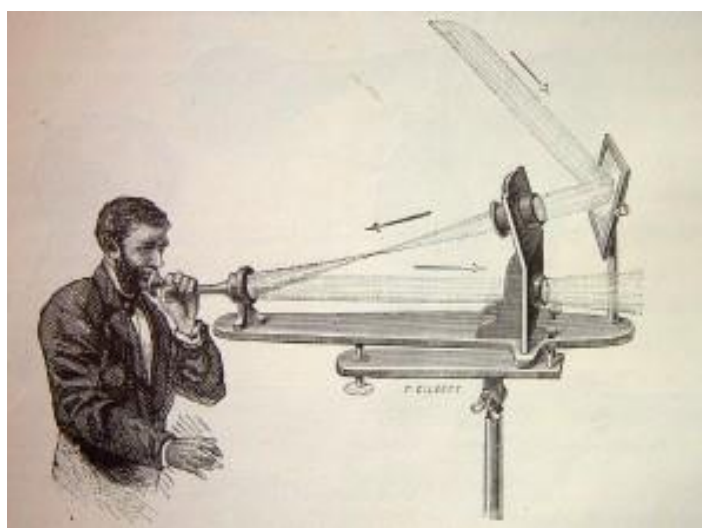
Nesse aspecto, sistemas de comunicação por luz visível (VLC) possuem uma importante vantagem em relação aos sistemas de telecomunicações mais utilizados atualmente, pois, ao contrário dos sinais de radiofrequência, a largura de banda dos sinais óticos é praticamente ilimitada e pode ser usada com licença livre no mundo todo (3).

Sistemas VLC tem mais flexibilidade e integridade do que outros sistemas de comunicação em alguns aspectos. Como o meio para transmissão em sistemas de VLC é a luz visível e não ondas de Radiofrequência (que podem penetrar nas paredes), a questão da segurança é inerentemente resolvida porque a luz não pode sair do recinto em que se encontra, que contém dados e informações em um único local.

Não há como recuperar e acessar essas informações a menos que haja um usuário no caminho direto da luz que está sendo usado para transmitir os dados. Além disso, os LEDs são altamente eficientes e estão se tornando mais duráveis ao longo dos anos, adicionando integridade a esses sistemas.

Esse sistema de comunicação tem a sua origem no ano de 1880 em Washington DC, quando o cientista escocês Graham Bell inventou o ‘fotofone’, dispositivo capaz de transmitir uma conversa telefônica por meio da luz a uma distância de centenas de metros. Tal dispositivo é considerado o primeiro equipamento de comunicação sem fio da história (4). A Figura 3 mostra o funcionamento de um fotofone.

Figura 3: Representação do funcionamento do fotofone



Fonte: POVEY, 2011



Devido ao grande sucesso dos sistemas de comunicação por fibra ótica e sistemas sem fio tais como Wi-Fi, o estudo relacionado ao VLC foi durante algum tempo postergado. No ano de 2003 no Japão foi estabelecido o Consórcio de Comunicação por Luz Visível (VLCC), que foi um consórcio entre grandes empresas japonesas para promover e padronizar a tecnologia VLC. A formação do VLCC estimulou interesse mundial no tema e o primeiro padrão IEEE para VLC – IEEE 802.15.7 – foi criado recentemente (5).

Em 2006, pesquisadores do CICTR no estado da Pensilvânia, EUA, propuseram um sistema de comunicação que utilizava uma combinação de PLC para transmitir informação por meio de cabos de energia até o local desejado e VLC para a comunicação *Indoor*. Esse método se mostrou bem promissor devido a sua alta eficiência e baixo custo (6).

Em 2010, uma equipe de pesquisadores da Siemens e do Heinrich Hertz Institute in Berlin realizaram transmissões à 500 Mbit/s com LEDs de cor branca há uma distância de 5 metros e transmissões em maiores distâncias com a taxa de 100 Mbit/s (7).

Em 2015, cientistas testaram a tecnologia VLC em escritórios e indústrias em Tallin, Estônia, e conseguiram obter velocidades superiores a 1 GB por segundo, o que é cem vezes superior à velocidade média do Wi-Fi atualmente (8).

Esses importantes avanços obtidos nessa área nos últimos anos demonstram que é necessário que se continue estudando VLC, pois essa tecnologia tem se mostrado bastante promissora e possivelmente será complementar ao Wi-Fi em alguns anos. Devido à necessidade de aprimorar os conhecimentos nesse tema, essa proposta de projeto de graduação foi realizada de modo a contribuir para a ampliação das informações já obtidas anteriormente. Com a montagem do protótipo, espera-se conferir a viabilidade do sistema para verificar se realmente essa tecnologia pode substituir ou interagir com as atuais presentes no mercado.

## **1.2 Possíveis Aplicações da Tecnologia VLC**

Existem situações nas quais o uso de tecnologias de rádio frequência para transmissão de dados é limitada para não causar interferência com os demais equipamentos presentes, como por exemplo hospitais e aviões. Além desses casos, ainda há casos em que o VLC pode ser útil de maneira a complementar a tecnologia já existente. Algumas dessas situações serão detalhadas abaixo.

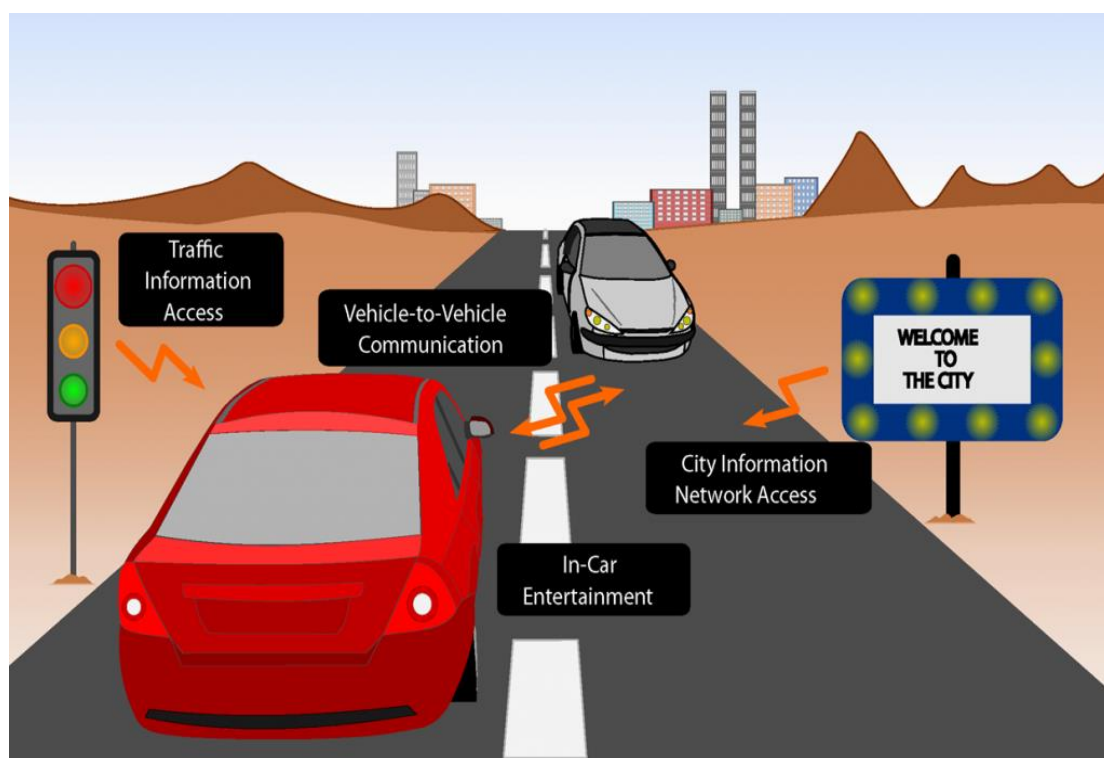
### **1.2.1 Troca de Informações no Trânsito de Veículos**

Em ruas e avenidas, semáforos utilizam a luz visível para realizar o controle do tráfego. Como os semáforos são muito comuns nas cidades, seria possível incorporar a eles um sistema de comunicação que permitisse que os motoristas e os pedestres que estivessem em estado de espera (aguardando o sinal verde) pudessem receber informações sobre o trânsito ou direções para uma determinada localização em seus aparelhos celulares. Dessa forma, se aproveitaria uma estrutura já existente para realizar uma comunicação de dados adicional.

Enquanto ter uma opção de conexão alternativa pode ser benéfico, essa opção poderia promover condutores a utilizarem celulares durante a condução, que pode ser perigoso. Se um motorista quiser acessar informação, isso deve ser feito de uma maneira que não cause danos ou perigo para o condutor ou outros motoristas. Uma maneira de fazer isso é incorporar um sistema de comunicação de luz visível no veículo e usar faróis do próprio veículo para enviar informações. Junto a isso, um sistema de comunicação de voz poderia ser implementado para que o motorista possa acessar informações sem tirar as mãos do volante.

Nesse cenário, um dos desafios a se contornar seria a luz ambiente em locais de alta incidência solar. Para tal, uma possível opção seria a utilização de fotodiodos nos receptores (celulares) que atenuam certos comprimentos de onda e são sensíveis aos demais comprimentos. A questão da quantidade de usuários permitidos simultaneamente é outro fator a ser analisado.

Figura 4: Comunicação no trânsito



Fonte: POVEY, 2011

### 1.2.2 Uso em Ambientes Hospitalares

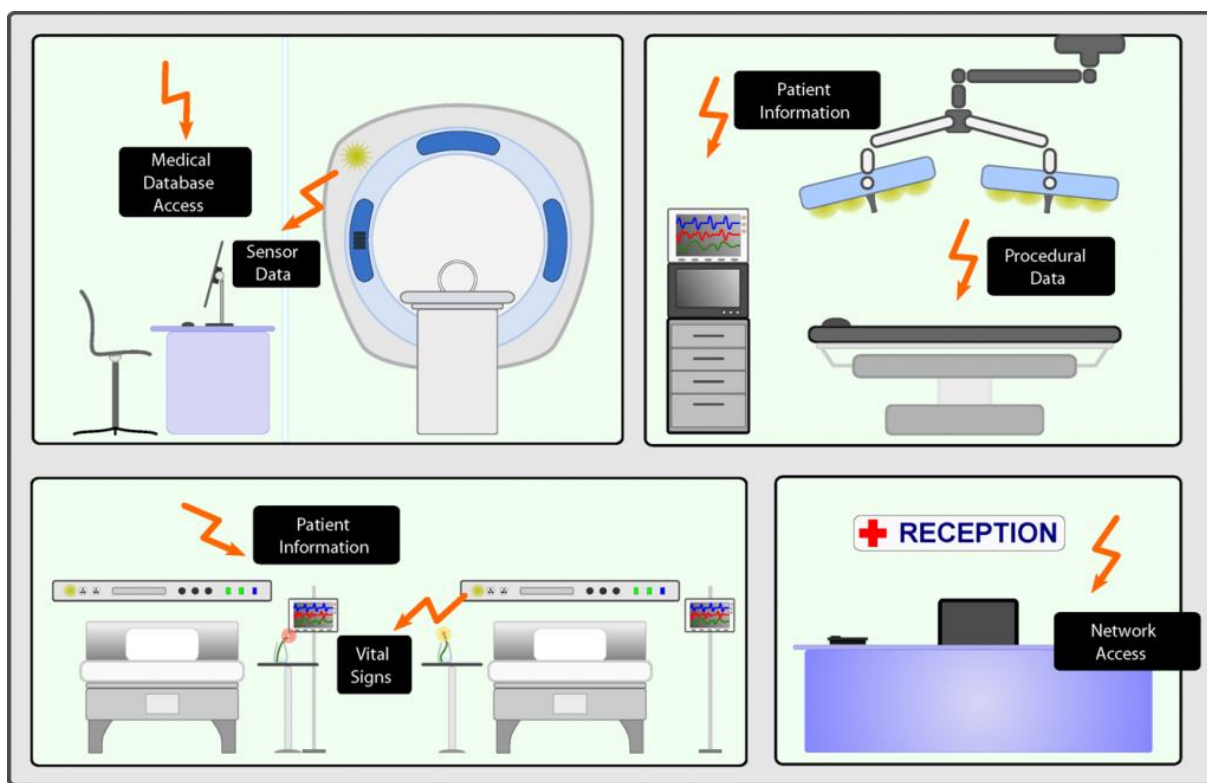
A questão da interferência eletromagnética entre equipamentos é, possivelmente, um dos fatores de maior atenção em ambientes hospitalares. Muitos equipamentos médico-hospitalares são sensíveis a distorções na forma de onda e uma possível interferência entre o sistema de comunicação sem fio e o instrumento médico pode causar uma perda de informação ou mesmo um “reset” do instrumento.

Nesse quesito, o VLC apresenta uma considerável vantagem em relação as tecnologias de radiofrequência, pois a troca de informações será apenas entre o emissor e o receptor que estarão posicionados de forma que os dados enviados atinjam somente o seu destino.

Uma possível forma de implementação seria utilizar as próprias lâmpadas fluorescentes, que são comuns em hospitais, para realizar a transmissão de dados. Dessa forma, se aproveitaria da mesma estrutura física tanto para iluminação quanto para trocas de informação.

A Figura 5 mostra uma possível forma de implementação de um sistema VLC em um hospital. Essa tecnologia pode ser utilizada para coletar dados de sensores, acessar uma base de dados médica, obter dados de pacientes ou mesmo para acesso à internet.

Figura 5: VLC em hospitais



Fonte: POVEY, 2011

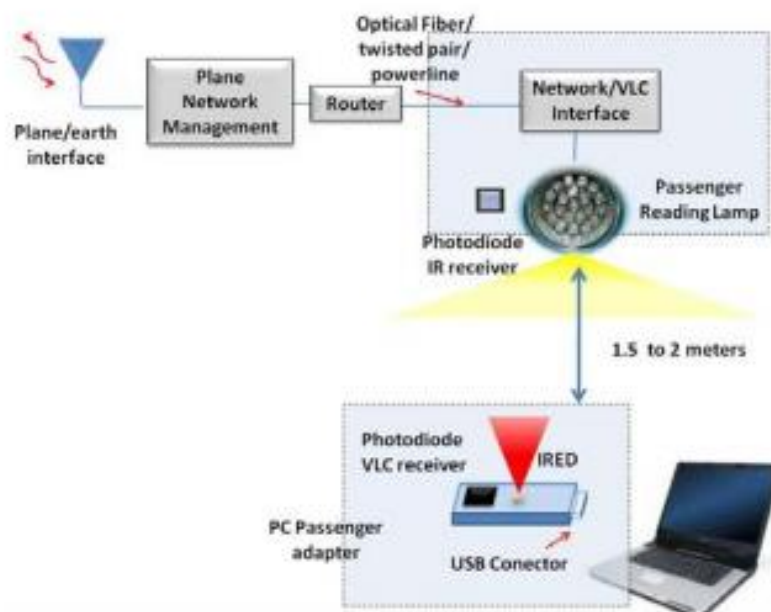
### 1.2.3 Acesso à Internet em Aeronaves

É de praxe em voos comerciais o aviso para que os passageiros desliguem seus aparelhos celulares ou os deixem em modo avião para não causar interferências com os demais equipamentos da aeronave. Em voos muito longos, ficar horas sem poder ter contato com “mundo exterior” pode ser prejudicial para pessoas que dependam de se comunicar constantemente para realizar negócios.

Já existem estudos na literatura (9) que mostram como implementar um sistema de Comunicação por Luz Visível em aeronaves de forma a fornecer uma rede de dados para passageiros durante voos. O diagrama de blocos do sistema proposto é mostrado a seguir.

Nesse sistema, a própria luz de leitura de cada passageiro é utilizada como o meio de transmissão de dados, além de fonte de iluminação.

Figura 6: Diagrama de blocos do sistema VLC em aviões



Fonte: PEREZ-JIMENEZ, 2011

## **2 OBJETIVOS GERAIS E ESPECÍFICOS**

Esse projeto tem por objetivo geral mostrar a viabilidade de sistemas de comunicação por luz visível e compará-lo com os demais sistemas de comunicação, mostrando suas vantagens e desvantagens em relação aos mais comuns presentes no mercado atual.

O objetivo específico consiste em projetar e montar um protótipo de sistema de comunicação por luz visível e, com esse protótipo, transmitir informação confiável de um ponto a outro a uma distância pequena (menos de 1m) e a uma velocidade compatível com a transmissão de dados.

Outra importante preocupação com o projeto foi o de montar o protótipo com componentes de baixo custo, mostrando a viabilidade tanto financeira quanto funcional do sistema de Comunicação de Luz Visível.

### 3 DESCRIÇÃO DO SISTEMA VLC IMPLEMENTADO

Este capítulo apresenta uma descrição dos elementos usados para a montagem do protótipo, além dos softwares utilizados para permitir que a mensagem transmitida no emissor pudesse ser recuperada na recepção.

#### 3.1 O Dispositivo de Conversão Eletro-Óptico

O transmissor de um sistema de comunicação é aquele responsável em enviar uma informação de um ponto a outro. No caso do sistema VLC, o dispositivo eletro-óptico pode ser formado por lâmpadas fluorescentes comuns para a transmissão de sinais a baixa velocidade (dezenas de kbit/s) ou por LEDs de alta luminosidade, que chegam à velocidade de envio de dados na casa das centenas de Mbit/s.

Para esse projeto, o LED utilizado foi o de cor branca de alto brilho (110 cd). A Figura 7 mostra os LEDs em funcionamento.

Figura 7: LED de cor branca utilizado no projeto

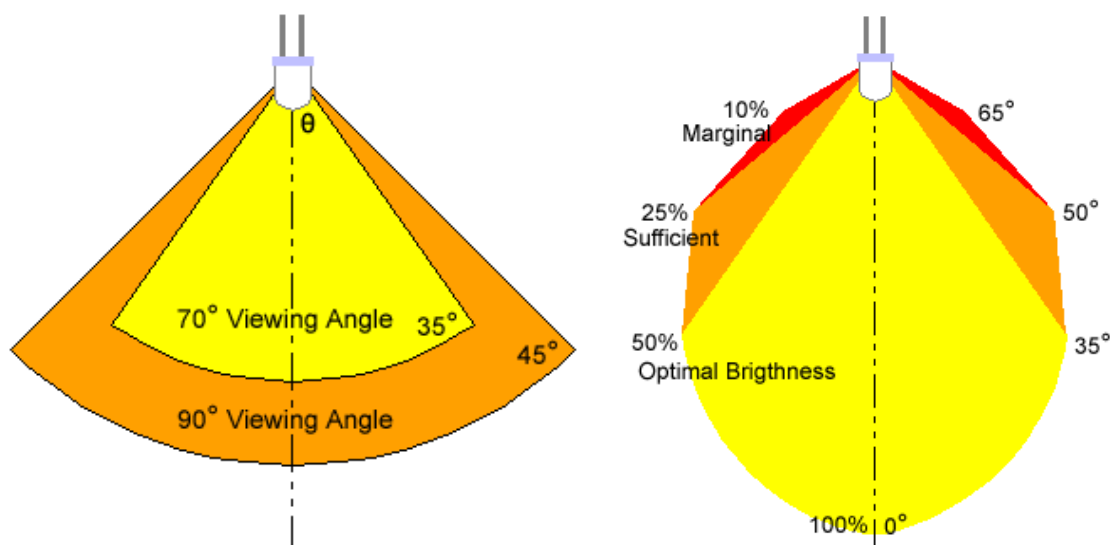


Fonte: Produção do próprio autor

Para medir a quantidade de luz emitida pelo LED, são utilizadas duas principais unidades de medida: o lúmen e a candela. Lumens está relacionado ao fluxo luminoso  $\Phi_v$ , que é a radiação emitida em todas as direções por uma fonte luminosa. Já a candela (cd) está relacionada com a intensidade luminosa  $I_v$ , que é a potência emitida por uma fonte de luz em uma determinada direção.

Outro fator importante para se levar em consideração para mensurar a qualidade do LED é o ângulo de visão. Esse ângulo é definido como o ângulo no qual o brilho do LED é metade do brilho para o ângulo igual a zero. Desse modo, se o ângulo  $\theta$  é o ângulo onde o brilho do LED é metade, então  $2\theta$  é o ângulo de visão cheia. A Figura 8 mostra essa situação.

Figura 8: Ângulo de visão de um LED



Fonte: GRANDWELL, 2007

Para que a maior quantidade possível de potência emitida pelo transmissor seja recebida pelo receptor, é necessário que esses dois estejam posicionados adequadamente de forma que os centros do LED e do fotodiodo estejam alinhados. A distância entre eles também é um fator determinante: se o emissor estiver muito afastado do receptor, é possível que parte da informação enviada seja perdida, pois o fotodiodo não seria capaz de capturar todos os dados enviados pelo transmissor.

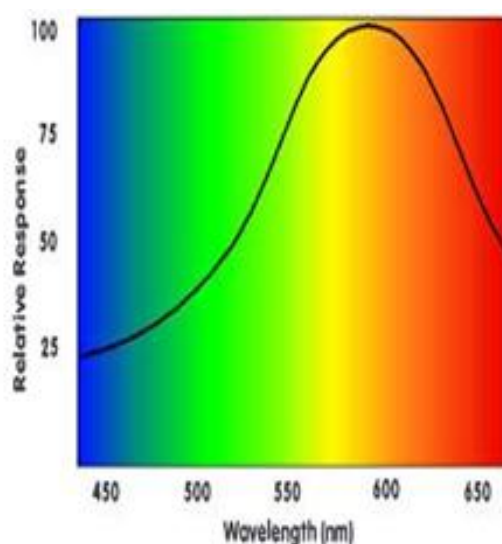


A cor emitida por um LED é definida por seu comprimento de onda, na qual a intensidade de radiação magnética é máxima  $\lambda_{max}$ . A relação entre  $\lambda_{max}$  e sua temperatura de cor T é definida pela lei do deslocamento de Wien:

$$\lambda_{max} = \frac{b}{T}, \quad (1)$$

onde b é a constante de dispersão de Wien em m.K (metro x Kelvin). O valor dessa constante é aproximadamente  $2,9 \times 10^{-3}$ m.K. Em geral, a região espectral de comprimento de onda de 565 a 600 nm é aquela em que sensibilidade do olho humano é a maior. Por isso, é mais fácil perceber variações de cores para LEDs amarelo e laranja. O gráfico 1 mostra essa relação.

Gráfico 1: Relação entre comprimento de onda e sensibilidade do olho humano



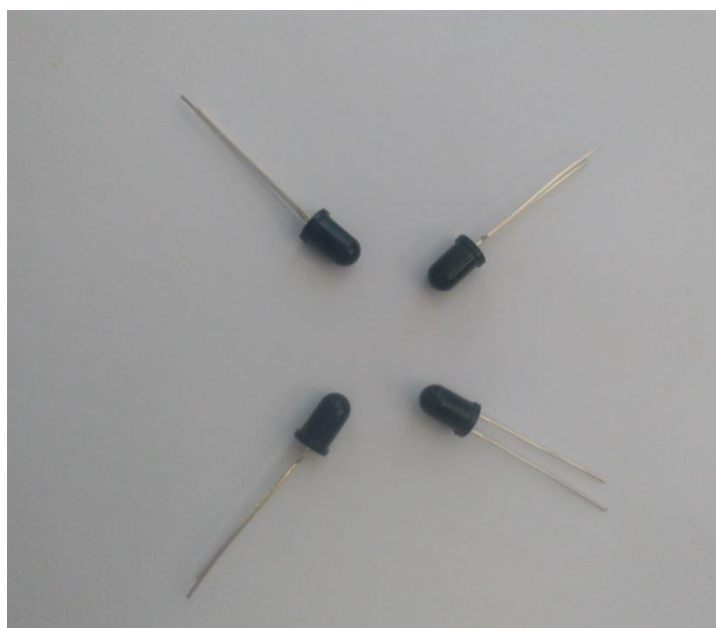
Fonte: GANAPATI, 2012

### 3.2 O Dispositivo de Conversão Óptico-Elétrico

Para que a mensagem enviada pelo emissor tenha alguma relevância, deve haver um receptor no sistema que receba e armazene a informação recebida. Por isso, fotodiodos são utilizados como receptores, pois eles reagem à luz emitida pelos LEDs e convertem essa luz em corrente.

Para esse projeto, devido a questões de custo, foram utilizados fotodiodos com o pico de sensibilidade no comprimento de onda de 940nm. Esse valor de pico não está no comprimento de luz visível (400 a 700 nm), no entanto atende aos requisitos do projeto se utilizado em conjunto com um amplificador operacional.

Figura 9: Fotodiodos utilizados no projeto

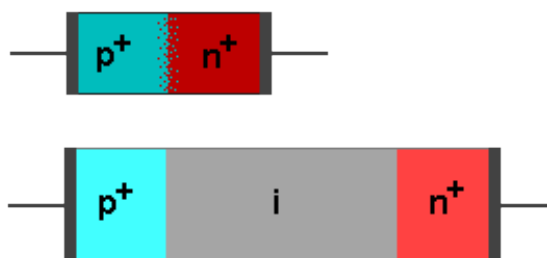


Fonte: Produção do próprio autor

Fotodiodos são similares a diodos semicondutores exceto pelo fato que estes podem ser tanto expostos ou fabricados com uma janela ou conexão de fibra ótica, de modo a permitir que a luz atinja a parte sensível do dispositivo.

A maioria dos diodos fabricados para funcionar como fotodiodos utiliza uma junção PIN ao invés de uma junção P-N, para aumentar a velocidade de resposta. Enquanto na junção P-N o dispositivo é formado por dois materiais do tipo P e do tipo N dopados no interior de um único cristal do semicondutor, na junção PIN existe uma camada I intrínseca sem dopagem altamente resistiva entre as camadas P e N. A Figura 10 mostra essa situação.

Figura 10: Representação de diodos do tipo P-N e PIN



Fonte: WOLFF, 2008

Quando um fóton com energia suficiente atinge o diodo, é criado um par elétron-buraco, também chamado de excíton, na rede cristalina do material semiconductor dopado. Essa região é caracterizada pela ausência de um elétron, se comportando como um portador de carga positiva. Esse efeito também é conhecido como efeito fotoelétrico.

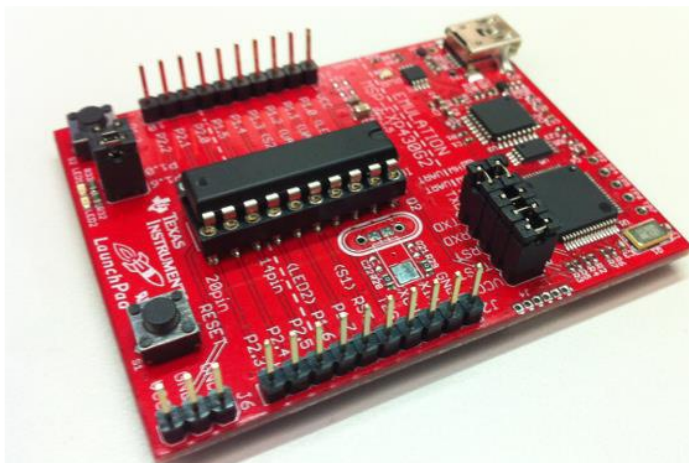
### 3.3 O Microcontrolador Launchpad MSP-EXP430G2 Usado como Conversor ADC

Como o sistema VLC transmite informação por meio de ondas de luz, o sinal analógico contínuo precisa ser discretizado para ser reconhecido pelo computador. Uma das formas de se fazer isso é com a utilização de um microcontrolador que faça a conversão analógica para digital através de um conversor ADC.

Um microcontrolador é um computador de tamanho reduzido em um único circuito integrado contendo um núcleo de processamento, memória e periféricos de entrada e saída programáveis. Os microcontroladores são projetados para aplicações embarcadas, em contraste com os microprocessadores que são utilizados em computadores pessoais ou outras aplicações de uso geral.

Para esse projeto, o microcontrolador escolhido foi o MSP430G2231, produzido pelo Texas Instruments Por ter Memória Flash e Memória RAM pequenas (2 Kb e 128b, respectivamente) algumas adaptações foram necessárias para compilar os códigos corretamente. A Figura 11 mostra o LaunchPad junto ao microcontrolador.

Figura 11: Launchpad MSP-EXP430G2

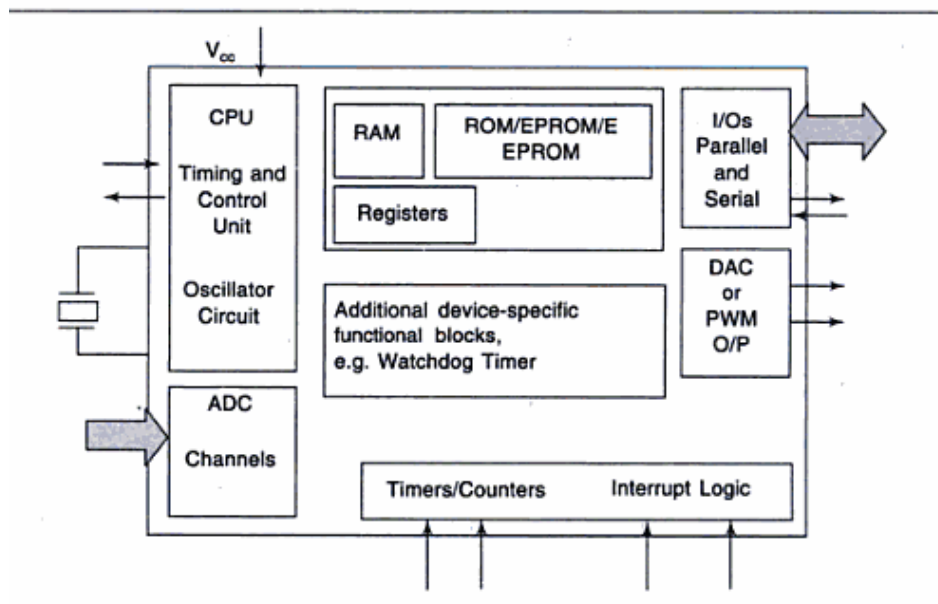


Fonte: Produção do próprio autor

Microcontroladores são usados em produtos e dispositivos controlados automaticamente, como os sistemas de controle de motor de automóvel, dispositivos médicos implantáveis, controles remotos, máquinas de escritório, eletrodomésticos, ferramentas elétricas, brinquedos e outros sistemas embarcados. Ao reduzir o tamanho e o custo em comparação a um projeto que usa um microprocessador, memória e de entrada / saída separados, microcontroladores fazem com que torne economicamente viável controlar digitalmente ainda mais dispositivos e processos.

O diagrama de blocos geral de um microcontrolador é mostrado na Figura 12. É importante observar que esse diagrama pode sofrer mudanças dependendo do dispositivo utilizado e também do fabricante do produto.

Figura 12: Diagrama de blocos de um microcontrolador



Fonte: DESHMUKH, 2005

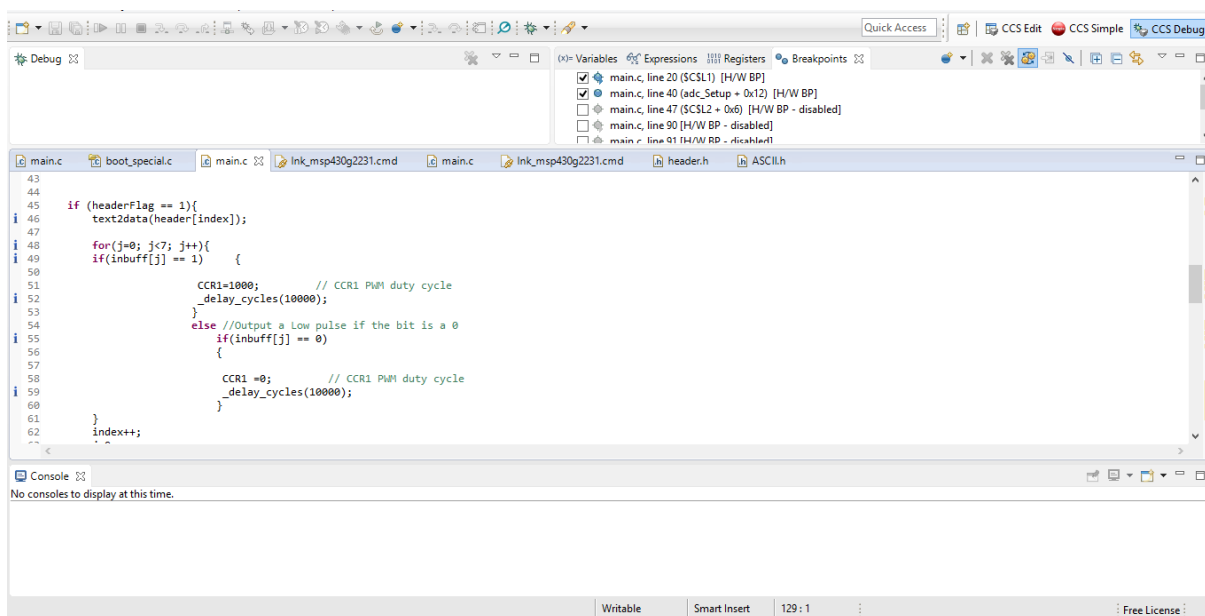
### 3.4 O Ambiente de Desenvolvimento para Microcontroladores

Code Composer Studio (CCS) é um ambiente integrado de desenvolvimento (IDE) para microcontroladores da Texas Instruments. Para este trabalho foi utilizado a versão 6.1.1 do software.

Com o auxílio deste software foi possível compilar e debugar os códigos que foram escritos para o emissor e receptor, além da utilização de pontos de quebra (breakpoints) para o envio de mensagens e também para salvar os dados coletados pelo receptor em um arquivo de texto. Esse arquivo de texto com os dados coletados foi posteriormente lido pelo Matlab, para fazer a conversão de volta da mensagem original.

A Figura 13 mostra uma parte do código do emissor na plataforma de debug do CCS, na qual é possível verificar os breakpoints ativos, os valores das variáveis utilizadas no programa, e também checar mensagens de erros ou avisos (*warnings*) que possam afetar o funcionamento correto do código

Figura 13: Interface do Code Composer Studio



Fonte: Produção do próprio autor

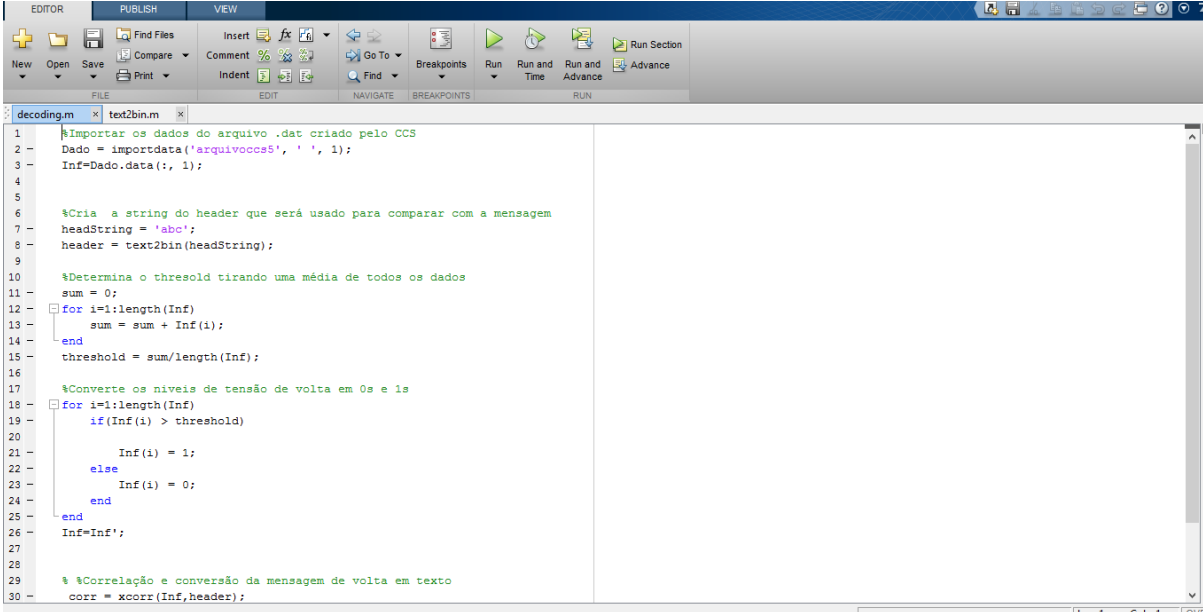
### 3.5 Leitura de Variáveis via Linguagem de Computação MATLAB

O MATLAB (MATrix LABoratory) foi o software utilizado para fazer a leitura dos valores salvos no arquivo texto pelo CCS e para converter esses valores de volta na mensagem original. Para tal, uma função que define o valor de *threshold* foi criada para converter os valores coletados no arquivo texto de volta em zeros e uns.

Feito essa conversão, outra função faz a correlação da mensagem coletada com a string do cabeçalho (header) da mensagem original. Após encontrada a sequência de zeros e uns que corresponde ao header da mensagem original, sabe-se que o que vem na sequência é a mensagem que se deseja obter.

A Figura 14 mostra a interface do software MATLAB com parte do código que faz a conversão da mensagem original. Os códigos completos do emissor e do receptor feitos no CCS e MATLAB podem ser verificados no apêndice no final deste trabalho.

Figura 14: Interface do MATLAB



```
1 %Importar os dados do arquivo .dat criado pelo CCS
2 Dado = importdata('arquivoccs5', ' ', 1);
3 Inf=Dado.data(:, 1);
4
5
6 %Cria a string do header que será usado para comparar com a mensagem
7 headString = 'abc';
8 header = text2bin(headString);
9
10 %Determina o threshold tirando uma média de todos os dados
11 sum = 0;
12 for i=1:length(Inf)
13     sum = sum + Inf(i);
14 end
15 threshold = sum/length(Inf);
16
17 %Converte os níveis de tensão de volta em 0s e 1s
18 for i=1:length(Inf)
19     if(Inf(i) > threshold)
20
21         Inf(i) = 1;
22     else
23         Inf(i) = 0;
24     end
25 end
26 Inf=Inf';
27
28
29 %Correlação e conversão da mensagem de volta em texto
30 corr = xcorr(Inf,header);
```

Fonte: Produção do próprio auto

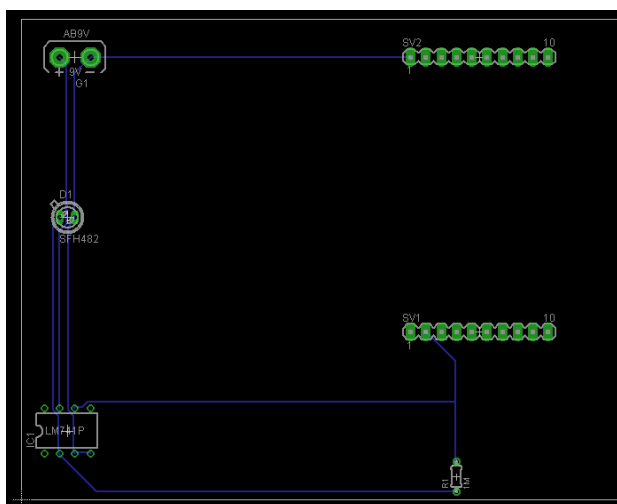
### 3.6 O Software usado no Projeto das Placas de Circuito Impresso

O EAGLE (Easy Applicable Graphical Layout Editor) é um software de projeto de Placas de Circuito Impresso (PCB) que foi utilizado para montar os circuitos utilizados no emissor e no receptor.

Depois de feito as ligações dos componentes utilizados, é possível organizá-los em um espaço correspondente a um espaço de um Placa de Circuito Impresso e utilizar um sistema de roteamento inteligente do software, que organiza as ligações dos componentes de forma a otimizar o espaço necessário para a montagem do circuito e utilizar o mínimo necessário de ligações externas (jumpers).

A Figura 15 mostra o roteamento do circuito do receptor deste projeto utilizando o comando de autoroute do EAGLE. Para não ser necessário fazer uma placa dupla face, a opção de usar duas camadas foi desmarcada.

Figura 15: circuito do receptor roteado pelo EAGLE



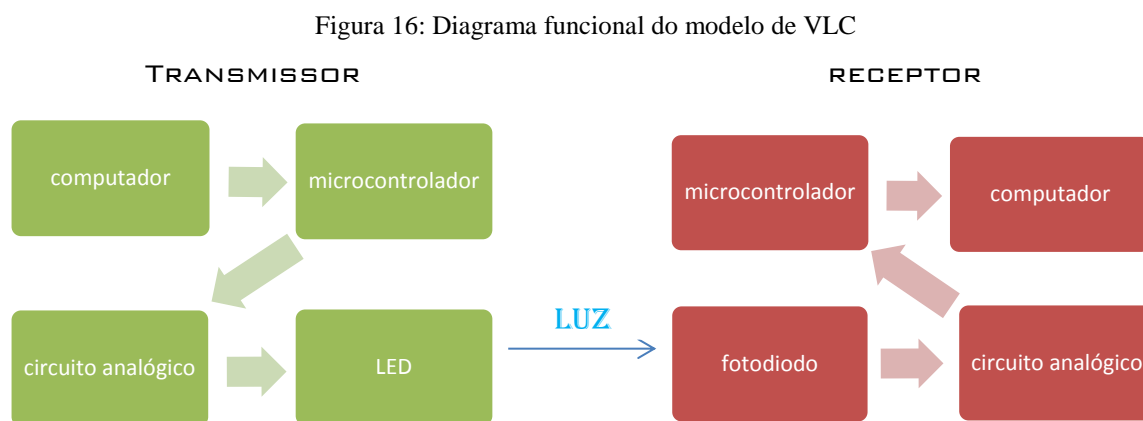
Fonte: Produção do próprio autor

No canto direito da Imagem 15 é possível ver os *pinheaders*, no qual será encaixado o Launchpad MSP430 no receptor. Com esse tipo de roteamento, foi possível fazer a placa em apenas uma camada, evitando maiores esforços na confecção da placa.



## 4 METODOLOGIA E DESENVOLVIMENTO DO PROJETO

A Figura 16 mostra o diagrama funcional do modelo de VLC desenvolvido neste projeto. O diagrama é dividido basicamente em duas partes: o transmissor contendo o dispositivo eletro-óptico e microcontrolador, e o receptor onde ocorre o processamento inverso.



Fonte: Produção do próprio autor

### 4.1 Descrição da Transmissão de Sinal VLC

No lado do transmissor, o computador produzirá a informação que deseja ser enviada, que para esse trabalho será uma mensagem de texto. No CCS, foram criados dois arquivos no formato .h (header). O primeiro deles possui as letras do cabeçalho e da mensagem que se deseja enviar, além do tamanho dos vetores relacionados a eles.

O segundo arquivo .h possui as respectivas correspondências dessas letras com sequências de zeros e uns de acordo com o código ASCII. Esses dois arquivos são incluídos em uma função main que converte esses valores de zeros e uns em níveis de tensão alto ou baixo por meio de uma função de Modulação por Largura de Pulso (PWM).

No caso de ser 1, a função PWM atribui uma largura de pulso de 100%. No caso de 0, a função PWM atribui uma largura de pulso de 0%. Esse controle da largura de pulso é feito

pelos registradores CCR0 e CCR1 utilizados no código do emissor, que está no apêndice deste trabalho. O registrador CCR0 define o período da modulação PWM, enquanto CCR1 define o ciclo de trabalho da onda.

Quanto mais próximo o valor CCR1 for de CCR0, maior será o ciclo de trabalho da modulação PWM. Dessa forma, para o bit 1, o valor de CCR1 é igual ao valor de CCR0 (100%), enquanto para o bit 0, o valor de CCR1=0 (0%).

O valor da frequência do PWM depende, além de CCR0, da fonte de clock utilizada. Para esse projeto foi utilizada o SMCLK, que é um clock de alta frequência. A frequência é dada pela divisão do SMCLK (por volta de 1,1Mhz) pelo valor de CCR0, que para esse projeto é utilizado o valor de 1000. Deste modo, obtém-se uma frequência próxima de 1,1Khz.

Esse valor de frequência necessitou ser reduzido pois a parte eletrônica do receptor não conseguia distinguir sequencias longas de zeros e uns para altas frequências e isso acabava comprometendo a conversão da mensagem original. Para contornar esse problema, são utilizadas as funções de *delay\_cycles*, que causam atraso no sinal, entretanto auxiliam para que o receptor consiga identificar mais precisamente o sinal enviado.

A Figura 17 mostra uma mensagem teste capturada na parte do emissor. Para essa mensagem, o *delay\_cycles* abaixou a frequência do PWM para aproximadamente 257 Hz. Como dito anteriormente, a mensagem é formada por um cabeçalho e a mensagem em si, seguindo o padrão da maioria dos protocolos de comunicação. Uma *flag* (headerflag) é usada para assinalar se o que está sendo transmitido é o cabeçalho ou a mensagem. No caso de headerflag = 1 significa que o cabeçalho está sendo convertido e enviado. No caso de headerflag=0 é a mensagem que é transmitida.

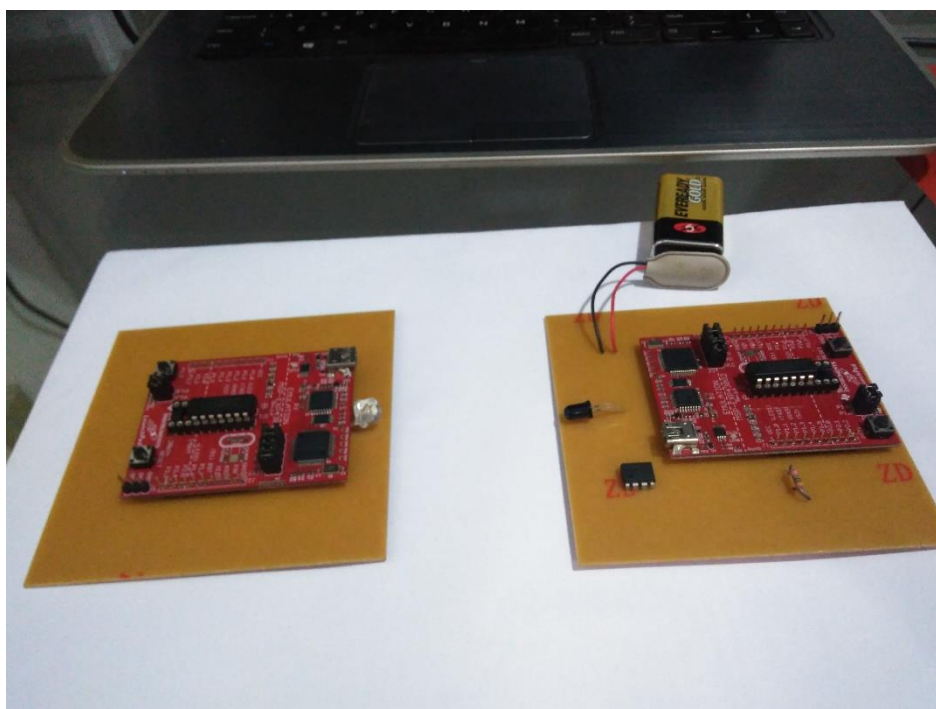


Devido a esse fato, os testes realizados nesse projeto foram limitados a distancias menores que 1m.

A questão da luz ambiente é outro ponto de atenção. Para lugares muito iluminados, o fotodiodo é excitado de forma que o sinal de saída no receptor é sempre de nível alto (como se recebesse um nível logico alto durante todo o tempo). Por isso, a luz ambiente também deve ser controlada para evitar tal problema.

A Figura 18 mostra o circuito eletrônico do emissor e do receptor. Para este projeto o Amp. Op. utilizado foi o LM741. Um resistor de  $1M\Omega$  foi ligado entre as portas 2 e 6 do amplificador para fazê-lo operar no modo multiplicador. A mesma porta 6 foi conectada ao LaunchPad do receptor para enviar o sinal amplificado.

Figura 18: Circuito eletrônico utilizado no projeto

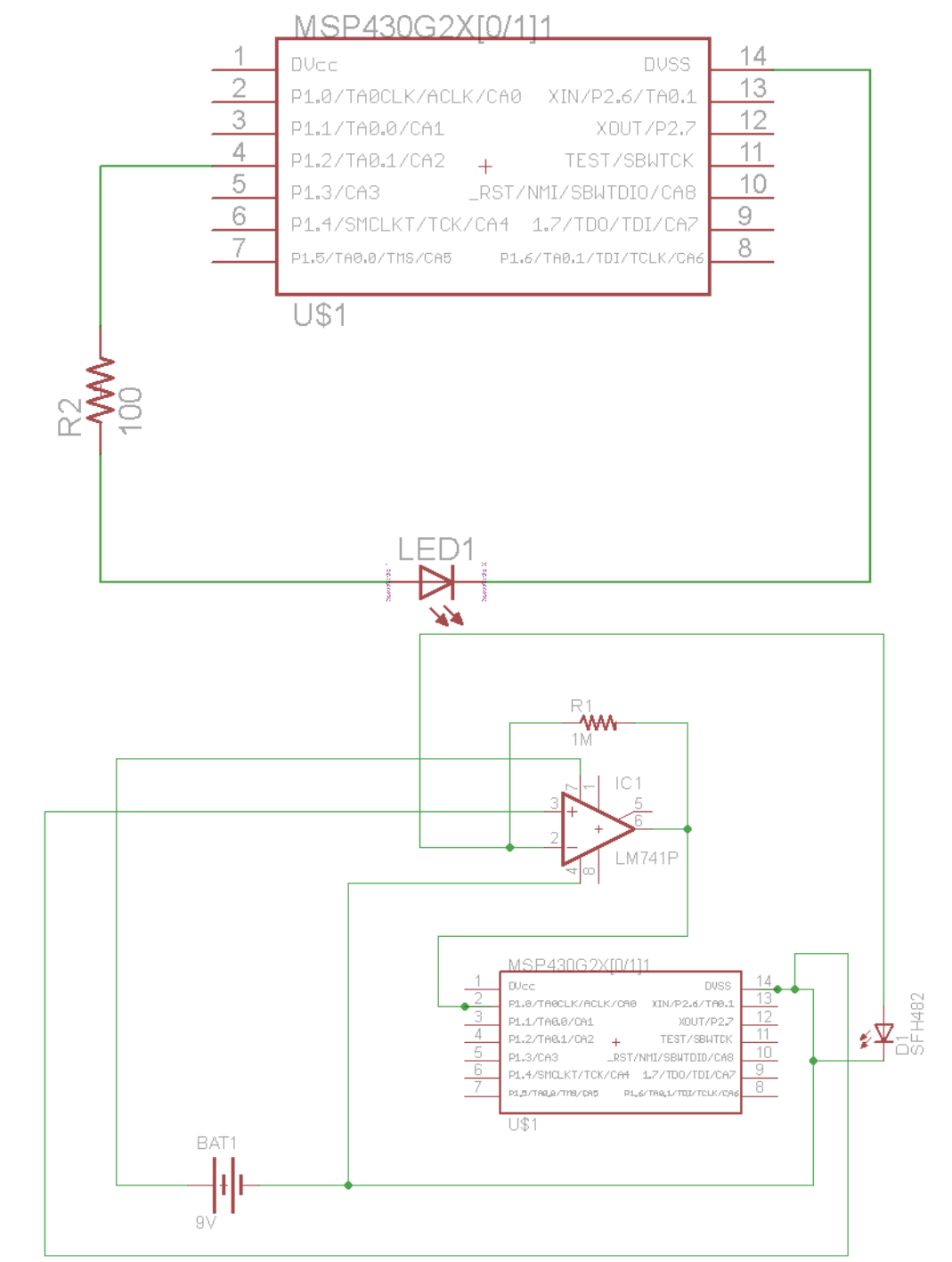


Fonte: Produção do próprio autor

Para a montagem do projeto na placa de circuito impresso (PCB) optou-se por montar um sistema do tipo Single Input – Single Output (SISO), com apenas um LED e um fotodiodo.

Utilizando o sistema *Multiple Input – Multiple Output* (MIMO) aumentava-se muito a complexidade do sistema e era mais difícil recuperar a informação original corretamente. A Figura 19 mostra a montagem dos circuitos do emissor e do receptor no EAGLE.

Figura 19: circuito do emissor e receptor no EAGLE



Fonte: Produção do próprio autor

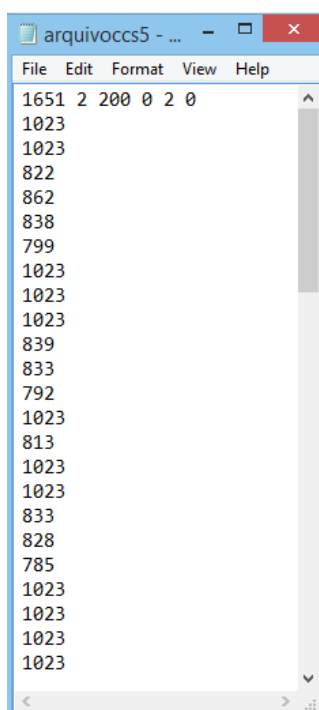
### 4.3 Descrição da Recepção de Sinal VLC

Após passar pelo circuito analógico, o sinal chega ao *Launchpad* do receptor. No código do CCS do receptor há duas importantes funções utilizadas para permitir a conversão do sinal. A primeira delas é uma função de setup, que define qual porta do MSP será utilizada como entrada, além de definir o número de conversões, habilitar o uso de interrupções e selecionar a opção de conversão analógica para digital (ADC) na porta P1.0.

A segunda função faz a conversão ADC das amostras coletadas. Para isso utiliza-se o registrador *Data Transfer Control* (DTC), que copia os valores da conversão em um vetor de 'n' posições definidas pelo usuário.

Um ponto de quebra (breakpoint) é utilizado quando a função de conversão ADC é chamada de forma que a cada valor convertido, esse valor é escrito em uma mensagem de texto. A Imagem 20 mostra os valores coletados e escritos na mensagem. Na primeira linha, há um cabeçalho criado pelo CCS, que será ignorado. Nas demais linhas, é possível ver os valores gravados.

Figura 20: arquivo texto os valores gravados do CCS



Fonte: Produção do próprio autor

Como a conversão ADC feita foi de 10 bits, o valor máximo registrável é de  $2^{10} - 1 = 1023$ , que representa o bit 1. Já para o bit 0, esse valor varia entre 750 e 900 para esse teste, mas pode alterar dependendo da quantidade de luz ambiente no momento da coleta.

O MATLAB importa os dados deste arquivo em um vetor e cria uma *string* correspondente ao header da mensagem original para poder fazer a comparação com os valores coletados. Essa *string* é convertida em valores binários por meio de uma função `text2bin` e, em seguida, é definido o *threshold* para converter a mensagem original também em valores binários.

Esse *threshold* é calculado a partir da média de todos os valores coletados. Se o valor coletado está abaixo dessa média, ele é considerado 0, se está acima é considerado 1. Finalmente, é utilizada a função `xcorr` do MATLAB para comparar o header que foi criado com a mensagem coletada.

Quando a função encontra na mensagem coletada a sequência de zeros e uns que correspondem ao cabeçalho, a outra função `bin2text` é utilizada para fazer a conversão dos bits seguintes ao cabeçalho, que correspondem à mensagem que se deseja obter.

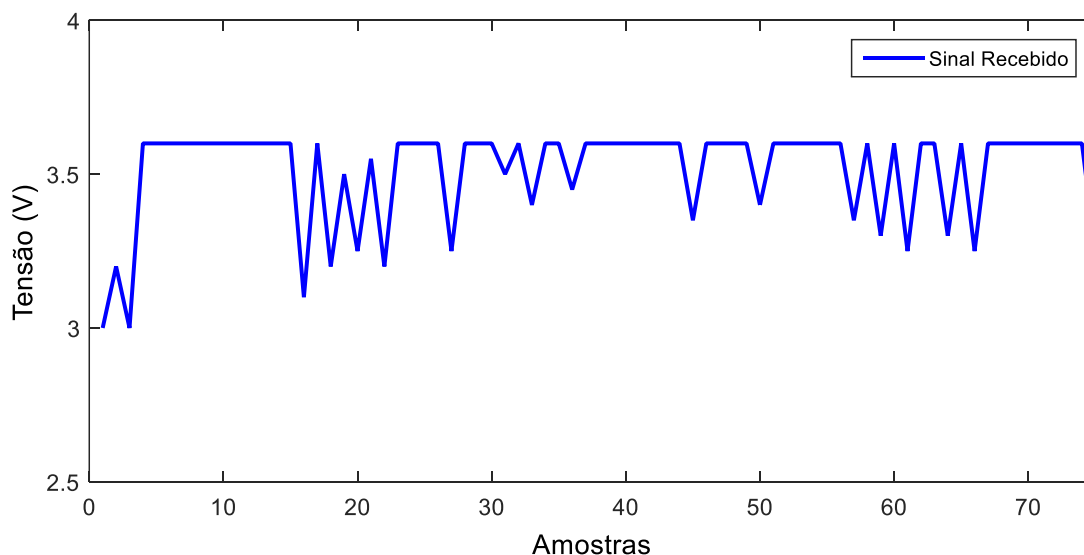
## 5 ANÁLISE DOS RESULTADOS DOS EXPERIMENTOS

Nesta seção, serão discutidos os resultados obtidos referentes às expectativas do projeto. Em linhas gerais, o resultado que esperava se alcançar no início do projeto, foi obtido com algumas limitações. Inicialmente esperava-se conseguir transmitir a mensagem com velocidades compatíveis com os atuais padrões de comunicação, porém alguns fatores impediram que isso ocorresse. O principal desses fatores foi o circuito analógico do receptor, que não conseguia distinguir sequencia longas de zeros e uns para frequências muito elevadas.

### 5.1 Avaliação do Efeito do Atraso na Transmissão

A Figura 21 mostra a mensagem reconstruída pelo receptor sem utilizar a função de “*delay cycles*” no código do emissor. Pela Figura, é possível notar que boa parte do conteúdo da mensagem foi perdido devido ao circuito analógico de receptor não identificar todos os bits da mensagem enviada pelo transmissor.

Figura 21: sinal distorcido no receptor



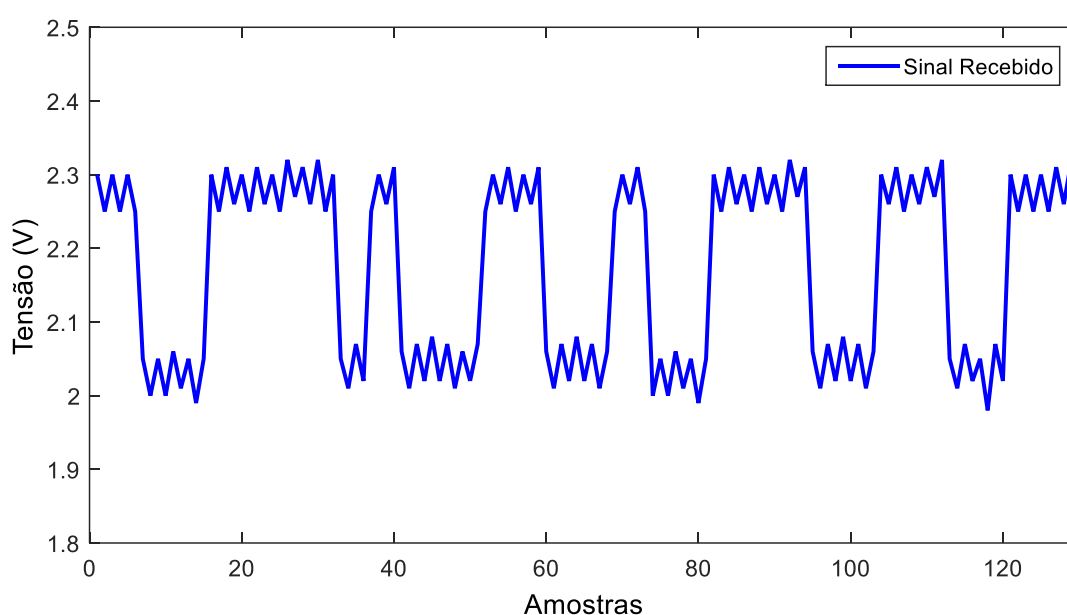
Fonte: Produção do próprio autor



## 5.2 Avaliação do Efeito da Distância de Comunicação

A questão da distância entre o emissor e o receptor foi um outro importante fator na transmissão de dados. A figura 22 mostra a forma de onda no receptor para uma distância de 90 cm do emissor para o receptor. Pela figura é possível notar que a diferença de nível de tensão entre o bit 0 e o bit 1 é bem pequeno (próximo de 0,3 V), além de aparecer um nível de ruído considerável na forma de onda.

Figura 22: sinal no receptor para uma distância de 90 cm



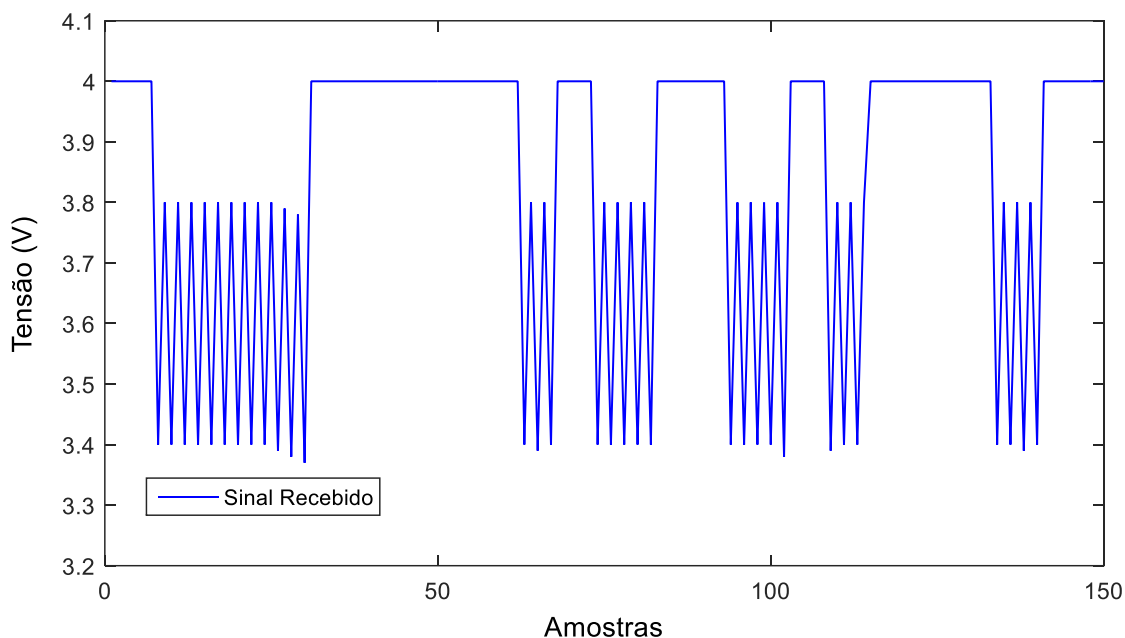
Fonte: Produção do próprio autor

## 5.3 Avaliação do Efeito do Ruído do Ambiente

Além da distância entre LED e fotodiodo, a luz ambiente foi uma outra questão que interferiu na transmissão de informação. A Figura 23 mostra a forma de onda no receptor em um ambiente bem iluminado, com o fotodiodo logo abaixo da lâmpada fluorescente.

Pela Figura é possível notar que para o nível lógico baixo, a forma de onda apresenta um elevado nível de ruído. Já para o nível lógico alto, a forma de onda satura, atingindo o valor próximo de 4 V.

Figura 23: sinal no receptor para um ambiente com alto índice de iluminação



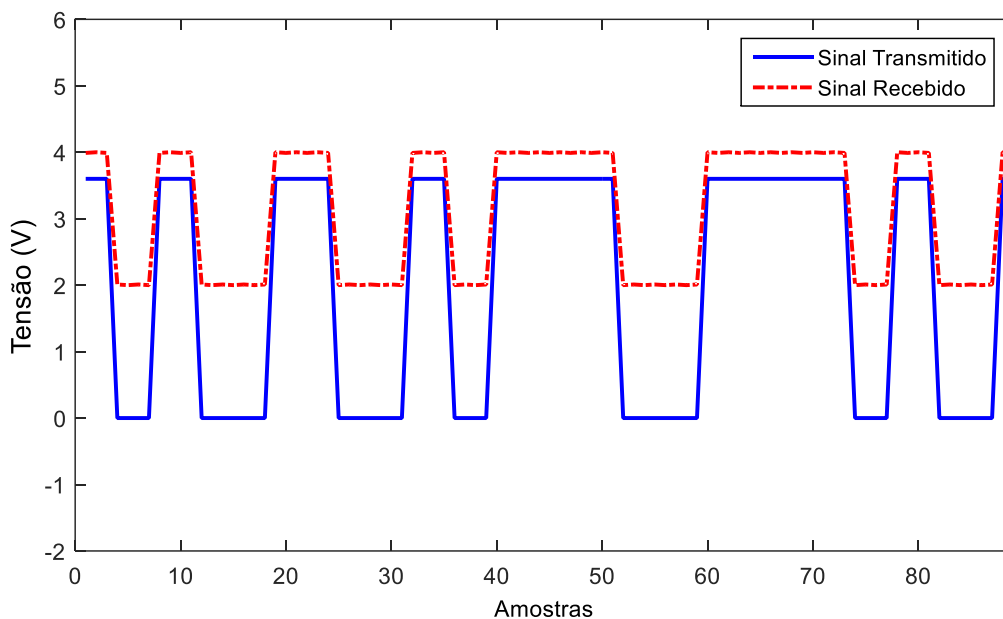
Fonte: Produção do próprio autor

Apesar da forma de onda sofrer considerável distorção nos dois últimos casos citados anteriormente, ainda foi possível fazer a conversão da mensagem original na maioria dos testes realizados. Controlando esses dois fatores externos, é possível obter formas de onda com maior precisão no receptor.

#### 5.4 Comunicação VLC em Condições Favoráveis

Na Figura 24 é possível observar o comparativo da forma de onda no emissor e receptor considerando as condições de distância do emissor e receptor e luz ambiente favoráveis (distância de 30 cm e pouca iluminação do ambiente). Nesse cenário, nota-se que a forma de onda no receptor se mostra muito próxima da forma de onda do emissor, apresentando apenas pequenos níveis de ruído e nível de tensão começando em 2V.

Figura 24: sinal no receptor para um ambiente com as condições favoráveis



Fonte: Produção do próprio autor

## 5.5 Análise das Limitações do Microcontrolador e da Coleta dos Dados

Outro fator que causou algumas limitações no projeto foi o microcontrolador utilizado. Por ter Memória Flash e Memória RAM pequenas (2 Kb e 128b, respectivamente), algumas adaptações foram necessárias para compilar os códigos corretamente. Na tabela ASCII, há no total 128 caracteres, contando os caracteres de controle, letras maiúsculas e minúsculas, símbolos e números.

Caso se optasse por escrever na memória do microcontrolador todos os valores binários correspondentes a cada um desses caracteres, ultrapassava-se o seu limite de memória. Devido a esse fato, a solução encontrada foi deixar comentado no código os caracteres não utilizados no envio da mensagem. Com isso, o limite da memória não foi excedido e a mensagem pode ser enviada com sucesso.

Tabela 1: tabela ASCII

# ASCII TABLE

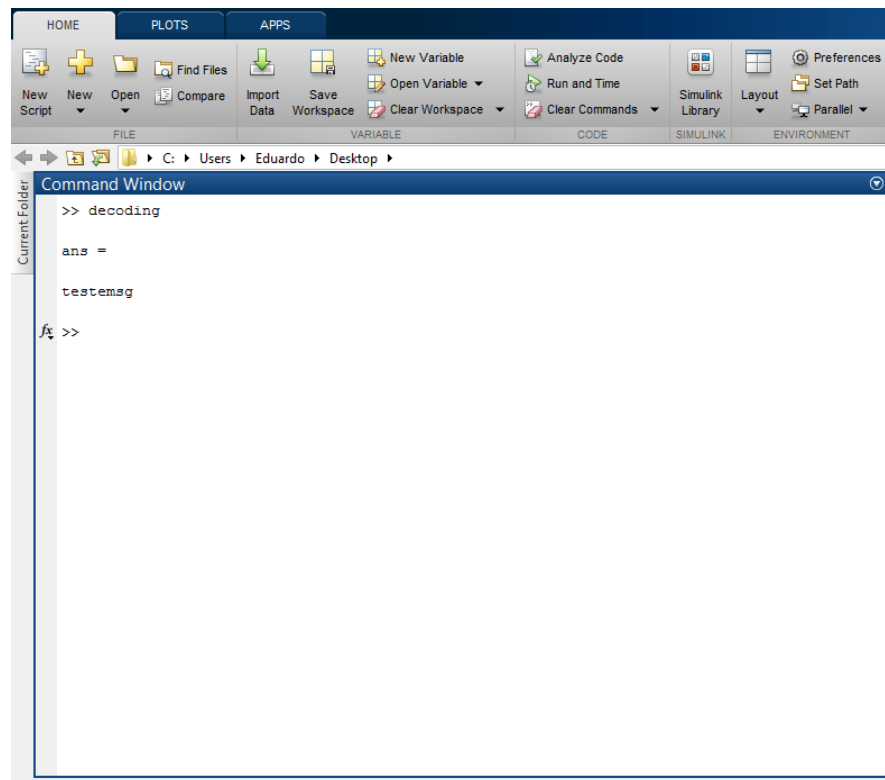
Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	-
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(	88	58	1011000	130	X					
41	29	101001	51	)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[					
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135	]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					

Fonte: ANONMOOS, 2010

Outra questão importante é referente a sincronia em que o *breakpoint* do receptor escreve os valores coletados no arquivo texto. Essa sincronia deve ocorrer de tal forma que na medida em que há mudança no bit transmitido, um novo valor seja escrito no arquivo. Para isso ocorrer, alguns ajustes na velocidade de transmissão tiveram que ser feitos com o intuito de que esse processo de escrita coincida com as mudanças do bit transmitido.

A Figura 25 mostra o resultado do envio de uma mensagem *'testemsg'* com *header* *'abc'*. Depois de encontrado a sequência de bits que representam o header, é feita a conversão dos demais bits para recuperar a mensagem original. Essa mensagem é exibida na janela de comando do MATLAB.

Figura 25: janela de comando do MATLAB com a mensagem enviada

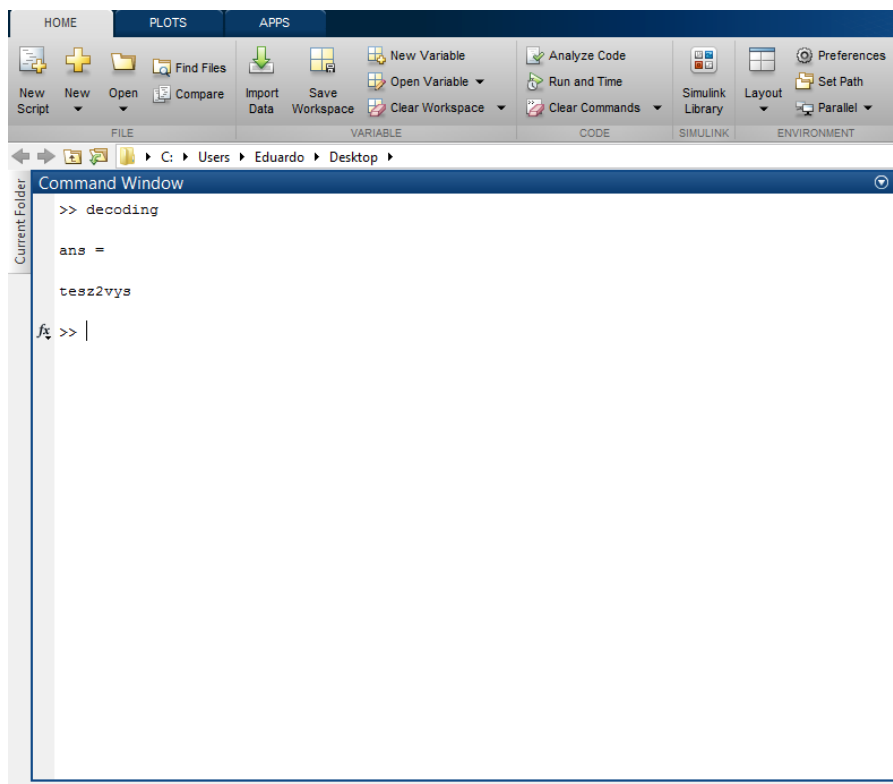


Fonte: Produção do próprio autor

Em alguns casos, quando se tinha longas sequências de zeros e uns, o processo de escrita dos valores do arquivo texto escrevia valores a mais do que o devido. No exemplo da mensagem 'testemsg', a sequência binária correspondente a letra 's' é  $s = \{1,1,1,0,0,1,1\}$  e a letra 't' é  $t = \{1,1,1,0,1,0,0\}$ , o que representa cinco uns seguidos na transição de 's' para 't'.

Em alguns testes realizados, ao invés de escrever os cinco valores correspondentes ao 1, o ponto de quebra do CCS escrevia seis valores, o que ocasionava erro na recuperação do sinal. A figura 26 mostra a mensagem recuperada nessa situação. Nota-se claramente que, nesse caso, apenas o início da mensagem foi corretamente recuperado. O restante, por haver um valor a mais escrito, ocasiona na recuperação de uma mensagem diferente da original.

Figura 26: janela de comando do MATLAB com a mensagem enviado com erro



Fonte: Produção do próprio autor

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Com esse trabalho foi possível compreender melhor o funcionamento de um sistema de Comunicação por Luz Visível e suas vantagens e desvantagens em relação aos sistemas de comunicação mais comuns atualmente.

Apesar de haver diversas situações em que o VLC possa ser utilizado, existem fatores que podem afetar o seu funcionamento. A questão da luz ambiente foi um dos fatores observados ao longo desse trabalho. Em ambientes muito iluminados, o circuito eletrônico do receptor teve dificuldades de captar o sinal enviado pelo emissor, o que ocasionou na recuperação de uma mensagem diferente da enviada.

Outro fator importante observado foi a distância entre os LEDs e os fotodiodos. Para distâncias próximas de 1m, já era possível perceber que parte do sinal enviado pelos LEDs começava a se perder na recepção. Utilizando componentes mais robustos, essa distância pode ser aumentada.

Como todo sistema de comunicação, VLC possui pontos positivos e negativos e, por isso, é necessário se fazer um estudo de caso para saber se essa tecnologia pode atender uma determinada solução esperada.

De modo geral, esse sistema de comunicação pode ser utilizado como uma forma de tecnologia complementar as que já estão consolidadas no mercado, mas ainda é necessário continuar o seu estudo para contornar suas principais limitações que foram citadas nesse trabalho.

Algumas possíveis melhorias podem ser realizadas para permitir maiores taxas de transmissão do protótipo, além do envio de outros tipos de informação, como por exemplo imagens ou áudios.

A primeira dessas possíveis melhorias é a utilização de um microcontrolador mais robusto, com maior espaço de memória Flash e RAM. Algumas adaptações tiveram que serem feitas

para os códigos rodarem adequadamente no MSP430G2231 utilizado que, de certa forma, ocasionaram certas limitações no envio de mensagens.

Outra possibilidade de melhoria é a utilização de fotodiodos com baixo tempo de resposta (na casa dos nanossegundos). Utilizando esses fotodiodos, possivelmente não seria mais necessário reduzir o tempo de envio de mensagem como feito nos testes descritos neste trabalho.

Uma outra possibilidade de melhoria é a transmissão de outros tipos de informação como voz e imagem. Para isso, seria necessário pensar em como implementar nesse atual sistema uma alternativa para permitir tal troca de informação.

No caso da imagem, como cada pixel é representado por um código binário que depende da cor que está se representando, poderia se pensar em uma maneira de converter esses pixels em sequências de zeros e uns para serem enviados pelo transmissor. O receptor capturaria esses dados e converteria de volta em pixels para reconstruir a imagem.



## 7 REFERÊNCIAS BIBLIOGRÁFICAS

- (1) SONG, J.K; LIM, J.S; KIM, H.M. A Modified A-law Companding Scheme for Indoor Visible Light Communication. **2014 IEEE fourth international conference on consumer electronics**, Berlim, p.254-255, set. 2014.
- (2) AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. **Plano de distribuição de faixas de frequências**. Disponível em: < <http://www.anatel.gov.br/grandeseventos/pt-br/espectro.html> >. Acesso em: 01 set. 2016.
- (3) MA, H.; LAMPE,L.; HRANILOVIC,S. Integration of Indoor Visible Light and Power Line Communication Systems. **2013 IEEE 17th International Symposium on Power Line Communications and Its Applications**, Joanesburgo, p. 291-296, mar. 2013.
- (4) POVEY,G. **The photophone – World’s first wireless communications**. Disponível em: <<http://visiblelightcomm.com/the-photophone-%E2%80%93-worlds-first-wireless-communications/>>. Acesso em: 01 set. 2015.
- (5)PUREVLC. **Differences between radio and Visible Light Communications**. Disponível em: < [http://www.purevlc.com/pureVLC\\_RadioTech\\_v1.0.pdf](http://www.purevlc.com/pureVLC_RadioTech_v1.0.pdf) > Acesso em: 02 set. 2016.
- (6) KAVREHAD, M., AMIRSHARI, P. Hybrid MV-LV power lines and white emitting diodes for triple-play broadband access communications. **IEC Comprehensive Report on Achieving the Triple Play: Technologies and Business Models for Success**, p.167-178 jan.2006.
- (7)SIEMENS. **500 Megabits/Second with White LED Light**. Disponível em: < <http://www.siemens.com/innovation/en/news/2010/500-megabits-second-with-white-led-light.htm>> Acesso em: 02 set. 2016.

(8) CREW, B. **Li-Fi has just been tested in the real world, and it's 100 times faster than Wi-Fi.** Disponível em: < <http://www.sciencealert.com/li-fi-tested-in-the-real-world-for-the-first-time-is-100-times-faster-than-wi-fi> > Acesso em: 22 set. 2016.

(9) PEREZ-JIMENEZ, R.; RUFO, J.; RABADAM, C.; LOPEZ-HERNANDEZ, F. Visible Light Communication Systems for Passenger In-Flight Data Networking. **2011 IEEE International Conference on Consumer Electronics (ICCE)**, p. 445-446, jan.2011.

(10) GUIMARÃES, R.; TOROBAY, D.; PEREIRA, E.; SEGATTO, M.; SILVA, J. Projeto e Avaliação de Desempenho de Sistemas DMT para Comunicação via Luz Visível. **MOMAG 2016**, jul. 2016.

## APÊNDICE

### 1.1 Códigos Code Composer Studio

```

//*****Emissor*****

#include "msp430G2231.h"
#include "ASCII.h"
#include "header.h"
#define LENGTHINBUFF 7
int inbuff[LENGTHINBUFF] = {0,0,0,0,0,0,0};
int index = 0;
int ii = 0;
int j=0;
int headerFlag = 1; //checa se está enviando o cabeçalho ou a mensagem
void text2data(char value);

void main(void)

{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT

    P1DIR |= BIT2; // P1.2 to output
    P1SEL |= BIT2; // P1.2 to TA0.1
    CCR0 = 1000-1; // PWM Period
    CCTL1 = OUTMOD_7; // CCR1 reset/set
    TACTL = TASSEL_2 + MC_1; // SMCLK, up mode

    while(1){

        //envia o cabeçalho primeiro

        if (headerFlag == 1){
            text2data(header[index]);

            for(j=0; j<7; j++){
                if(inbuff[j] == 1) { ////Envia um pulso alto se o bit for 1

                    CCR1=1000; // CCR1 PWM duty cycle
                    _delay_cycles(100);
                }
                else //Envia um pulso baixo se o bit for 0
                if(inbuff[j] == 0){

                    CCR1 = 0; // CCR1 PWM duty cycle
                    _delay_cycles(100);
                }
            }
            index++;
            j=0;

            if (index == HEADSIZE){ //end of header

```

```

        index = 0;
        headerFlag = 0; //headerflag=0, mensagem começa a ser transmitida
        }          }

// envia a mensagem

    else if (headerFlag == 0){
        text2data(msg[index]);

        for(j=0; j<7; j++){
            if(inbuff[j] == 1)    { ////Envia um pulso alto se o bit for 1

                CCR1=1000;        // CCR1 PWM duty cycle
                _delay_cycles(100);}
            else
                if(inbuff[j] == 0) {//Envia um pulso baixo se o bit for 0

                    CCR1 =0;        // CCR1 PWM duty cycle
                    _delay_cycles(100);
                }
            index++;
            j=0;

            if (index == MSGSIZE){        //final da mensagem
                index = 0;
                headerFlag = 1;        //mensagem termina de ser transmitida
            }
        }
    }
}

```

// funcao que converte os caracteres em sequencia de zeros e uns de acordo com o codigo ASCII

```

void text2data(char value){
    switch (value){
        case 'a':
            for(ii=0; ii<7; ii++)
                inbuff[ii] = a[ii];
            break;
        case 'b':
            for(ii=0; ii<7; ii++)
                inbuff[ii] = b[ii];
            break;
        case 'c':
            for(ii=0; ii<7; ii++)
                inbuff[ii] = c[ii];
            break;
        case 'd':
            for(ii=0; ii<7; ii++)
                inbuff[ii] = d[ii];
            break;
        case 'e':
            for(ii=0; ii<7; ii++)
                inbuff[ii] = e[ii];
            break;
    }
}

```

```
case 'f':
for(ii=0; ii<7; ii++)
inbuff[ii] = f[ii];
break;
case 'g':
for(ii=0; ii<7; ii++)
inbuff[ii] = g[ii];
break;
case 'h':
for(ii=0; ii<7; ii++)
inbuff[ii] = h[ii];
break;
case 'i':
for(ii=0; ii<7; ii++)
inbuff[ii] = i[ii];
break;
case 'j':
for(ii=0; ii<7; ii++)
inbuff[ii] = j[ii];
break;
case 'k':
for(ii=0; ii<7; ii++)
inbuff[ii] = k[ii];
break;
case 'l':
for(ii=0; ii<7; ii++)
inbuff[ii] = l[ii];
break;
case 'm':
for(ii=0; ii<7; ii++)
inbuff[ii] = m[ii];
break;
case 'n':
for(ii=0; ii<7; ii++)
inbuff[ii] = n[ii];
break;
case 'o':
for(ii=0; ii<7; ii++)
inbuff[ii] = o[ii];
break;
case 'p':
for(ii=0; ii<7; ii++)
inbuff[ii] = p[ii];
break;
case 'q':
for(ii=0; ii<7; ii++)
inbuff[ii] = q[ii];
break;
case 'r':
for(ii=0; ii<7; ii++)
inbuff[ii] = r[ii];
break;
case 's':
for(ii=0; ii<7; ii++)
inbuff[ii] = s[ii];
break;
case 't':
for(ii=0; ii<7; ii++)
```

```
inbuff[ii] = t[ii];
break;
case 'u':
for(ii=0; ii<7; ii++)
inbuff[ii] = u[ii];
break;
case 'v':
for(ii=0; ii<7; ii++)
inbuff[ii] = v[ii];
break;
case 'w':
for(ii=0; ii<7; ii++)
inbuff[ii] = w[ii];
break;
case 'x':
for(ii=0; ii<7; ii++)
inbuff[ii] = x[ii];
break;
case 'y':
for(ii=0; ii<7; ii++)
inbuff[ii] = y[ii];
break;
case 'z':
for(ii=0; ii<7; ii++)
inbuff[ii] = z[ii];
break;
/*case 'A':
for(ii=0; ii<7; ii++)
inbuff[ii] = A[ii];
break;
case 'B':
for(ii=0; ii<7; ii++)
inbuff[ii] = B[ii];
break;
case 'C':
for(ii=0; ii<7; ii++)
inbuff[ii] = C[ii];
break;
case 'D':
for(ii=0; ii<7; ii++)
inbuff[ii] = D[ii];
break;
case 'E':
for(ii=0; ii<7; ii++)
inbuff[ii] = E[ii];
break;
case 'F':
for(ii=0; ii<7; ii++)
inbuff[ii] = F[ii];
break;
case 'G':
for(ii=0; ii<7; ii++)
inbuff[ii] = G[ii];
break;
case 'H':
for(ii=0; ii<7; ii++)
inbuff[ii] = H[ii];
break;
```

```
case 'I':
for(ii=0; ii<7; ii++)
inbuff[ii] = I[ii];
break;
case 'J':
for(ii=0; ii<7; ii++)
inbuff[ii] = J[ii];
break;
case 'K':
for(ii=0; ii<7; ii++)
inbuff[ii] = K[ii];
break;
case 'L':
for(ii=0; ii<7; ii++)
inbuff[ii] = L[ii];
break;
case 'M':
for(ii=0; ii<7; ii++)
inbuff[ii] = M[ii];
break;
case 'N':
for(ii=0; ii<7; ii++)
inbuff[ii] = N[ii];
break;
case 'O':
for(ii=0; ii<7; ii++)
inbuff[ii] = O[ii];
break;
case 'P':
for(ii=0; ii<7; ii++)
inbuff[ii] = P[ii];
break;
case 'Q':
for(ii=0; ii<7; ii++)
inbuff[ii] = Q[ii];
break;
case 'R':
for(ii=0; ii<7; ii++)
inbuff[ii] = R[ii];
break;
case 'S':
for(ii=0; ii<7; ii++)
inbuff[ii] = S[ii];
break;
case 'T':
for(ii=0; ii<7; ii++)
inbuff[ii] = T[ii];
break;
case 'U':
for(ii=0; ii<7; ii++)
inbuff[ii] = U[ii];
break;
case 'V':
for(ii=0; ii<7; ii++)
inbuff[ii] = V[ii];
break;
case 'W':
for(ii=0; ii<7; ii++)
```

```
inbuff[ii] = W[ii];
break;
case 'X':
for(ii=0; ii<7; ii++)
inbuff[ii] = X[ii];
break;
case 'Y':
for(ii=0; ii<7; ii++)
inbuff[ii] = Y[ii];
break;
case 'Z':
for(ii=0; ii<7; ii++)
inbuff[ii] = Z[ii];
break;
case '0':
for(ii=0; ii<7; ii++)
inbuff[ii] = zero[ii];
break;
case '1':
for(ii=0; ii<7; ii++)
inbuff[ii] = one[ii];
break;
case '2':
for(ii=0; ii<7; ii++)
inbuff[ii] = two[ii];
break;
case '3':
for(ii=0; ii<7; ii++)
inbuff[ii] = three[ii];
break;
case '4':
for(ii=0; ii<7; ii++)
inbuff[ii] = four[ii];
break;
case '5':
for(ii=0; ii<7; ii++)
inbuff[ii] = five[ii];
break;
case '6':
for(ii=0; ii<7; ii++)
inbuff[ii] = six[ii];
break;
case '7':
for(ii=0; ii<7; ii++)
inbuff[ii] = seven[ii];
break;
case '8':
for(ii=0; ii<7; ii++)
inbuff[ii] = eight[ii];
break;
case '9':
for(ii=0; ii<7; ii++)
inbuff[ii] = nine[ii];
break;
case '!':
for(ii=0; ii<7; ii++)
inbuff[ii] = excl[ii];
break;
```



```
case '":
for(ii=0; ii<7; ii++)
inbuff[ii] = quot[ii];
break;
case '#':
for(ii=0; ii<7; ii++)
inbuff[ii] = hash[ii];
break;
case '$':
for(ii=0; ii<7; ii++)
inbuff[ii] = dola[ii];
break;
case '%':
for(ii=0; ii<7; ii++)
inbuff[ii] = perc[ii];
break;
case '&':
for(ii=0; ii<7; ii++)
inbuff[ii] = amp[ii];
break;
case '(':
for(ii=0; ii<7; ii++)
inbuff[ii] = lpar[ii];
break;
case ')':
for(ii=0; ii<7; ii++)
inbuff[ii] = rpar[ii];
break;
case '*':
for(ii=0; ii<7; ii++)
inbuff[ii] = time[ii];
break;
case '+':
for(ii=0; ii<7; ii++)
inbuff[ii] = add[ii];
break;
case ',':
for(ii=0; ii<7; ii++)
inbuff[ii] = comm[ii];
break;
case '-':
for(ii=0; ii<7; ii++)
inbuff[ii] = dash[ii];
break;
case '.':
for(ii=0; ii<7; ii++)
inbuff[ii] = peri[ii];
break;
case '/':
for(ii=0; ii<7; ii++)
inbuff[ii] = slsh[ii];
break;
case ':':
for(ii=0; ii<7; ii++)
inbuff[ii] = col[ii];
break;
case ';':
for(ii=0; ii<7; ii++)
```

```

inbuff[ii] = scol[ii];
break;
case '<':
for(ii=0; ii<7; ii++)
inbuff[ii] = lthn[ii];
break;
case '=':
for(ii=0; ii<7; ii++)
inbuff[ii] = equl[ii];
break;
case '>':
for(ii=0; ii<7; ii++)
inbuff[ii] = gthn[ii];
break;
case '?':
for(ii=0; ii<7; ii++)
inbuff[ii] = ques[ii];
break;
case '@':
for(ii=0; ii<7; ii++)
inbuff[ii] = at[ii];
break;
case '{':
for(ii=0; ii<7; ii++)
inbuff[ii] = lcbk[ii];
break;
case '}':
for(ii=0; ii<7; ii++)
inbuff[ii] = rcbk[ii];
break;
case '~':
for(ii=0; ii<7; ii++)
inbuff[ii] = til[ii];
break;
case ' ':
for(ii=0; ii<7; ii++)
inbuff[ii] = spac[ii];
break;
default:
for(ii=0; ii<7; ii++)
inbuff[ii] = 0;
break;

```

```

} }

```

```

/* arquivo header .h com cabeçalho e mensagem que serão enviados */
const char msg[8] = {'t','e','s','t','e','m','s','g'};
//const char msg[5]= {'t','e','s','t','e'}
const char header[3] = {'a','b','c'};
//const char header[6] = {'h','e','a','d','e','r'};
const int MSGSIZE = 8;
const int HEADSIZE = 3;

```

```
/* * Arquivo ASCII.h que define os valores binarios de cada caracter no codigo
ASCII */
```

```
/*Numeros
```

```
const int zero[8] = {0,1,1,0,0,0,0};
const int one[8] = {0,1,1,0,0,0,1};
const int two[8] = {0,1,1,0,0,1,0};
const int three[8]= {0,1,1,0,0,1,1};
const int four[8] = {0,1,1,0,1,0,0};
const int five[8] = {0,1,1,0,1,0,1};
const int six[8] = {0,1,1,0,1,1,0};
const int seven[8]= {0,1,1,0,1,1,1};
const int eight[8]= {0,1,1,1,0,0,0};
const int nine[8] = {0,1,1,1,0,0,1}; */
```

```
//Letras minusculas
```

```
const int a[8] = {1,1,0,0,0,0,1};
const int b[8] = {1,1,0,0,0,1,0};
const int c[8] = {1,1,0,0,0,1,1};
const int d[8] = {1,1,0,0,1,0,0};
const int e[8] = {1,1,0,0,1,0,1};
const int f[8] = {1,1,0,0,1,1,0};
const int g[8] = {1,1,0,0,1,1,1};
const int h[8] = {1,1,0,1,0,0,0};
const int i[8] = {1,1,0,1,0,0,1};
const int j[8] = {1,1,0,1,0,1,0};
const int k[8] = {1,1,0,1,0,1,1};
const int l[8] = {1,1,0,1,1,0,0};
const int m[8] = {1,1,0,1,1,0,1};
const int n[8] = {1,1,0,1,1,1,0};
const int o[8] = {1,1,0,1,1,1,1};
const int p[8] = {1,1,1,0,0,0,0};
const int q[8] = {1,1,1,0,0,0,1};
const int r[8] = {1,1,1,0,0,1,0};
const int s[8] = {1,1,1,0,0,1,1};
const int t[8] = {1,1,1,0,1,0,0};
const int u[8] = {1,1,1,0,1,0,1};
const int v[8] = {1,1,1,0,1,1,0};
const int w[8] = {1,1,1,0,1,1,1};
const int x[8] = {1,1,1,1,0,0,0};
const int y[8] = {1,1,1,1,0,0,1};
const int z[8] = {1,1,1,1,0,1,0};
```

```
/*Letras maiusculas
```

```
const int A[8] = {1,0,0,0,0,0,1};
const int B[8] = {1,0,0,0,0,1,0};
const int C[8] = {1,0,0,0,0,1,1};
const int D[8] = {1,0,0,0,1,0,0};
const int E[8] = {1,0,0,0,1,0,1};
const int F[8] = {1,0,0,0,1,1,0};
const int G[8] = {1,0,0,0,1,1,1};
const int H[8] = {1,0,0,1,0,0,0};
const int I[8] = {1,0,0,1,0,0,1};
const int J[8] = {1,0,0,1,0,1,0};
const int K[8] = {1,0,0,1,0,1,1};
const int L[8] = {1,0,0,1,1,0,0};
const int M[8] = {1,0,0,1,1,0,1};
const int N[8] = {1,0,0,1,1,1,0};
```

```

const int O[8] = {1,0,0,1,1,1,1};
const int P[8] = {1,0,1,0,0,0,0};
const int Q[8] = {1,0,1,0,0,0,1};
const int R[8] = {1,0,1,0,0,1,0};
const int S[8] = {1,0,1,0,0,1,1};
const int T[8] = {1,0,1,0,1,0,0};
const int U[8] = {1,0,1,0,1,0,1};
const int V[8] = {1,0,1,0,1,1,0};
const int W[8] = {1,0,1,0,1,1,1};
const int X[8] = {1,0,1,1,0,0,0};
const int Y[8] = {1,0,1,1,0,0,1};
const int Z[8] = {1,0,1,1,0,1,0};

//Simbolos
const int excl[8] = {0,1,0,0,0,0,1};
const int quot[8] = {0,1,0,0,0,1,0};
const int hash[8] = {0,1,0,0,0,1,1};
const int dola[8] = {0,1,0,0,1,0,0};
const int perc[8] = {0,1,0,0,1,0,1};
const int amp[8] = {0,1,0,0,1,1,0};
const int apst[8] = {0,1,0,0,1,1,1};
const int lpar[8] = {0,1,0,1,0,0,0};
const int rpar[8] = {0,1,0,1,0,0,1};
const int time[8] = {0,1,0,1,0,1,0};
const int add[8] = {0,1,0,1,0,1,1};
const int comm[8] = {0,1,0,1,1,0,0};
const int dash[8] = {0,1,0,1,1,0,1};
const int peri[8] = {0,1,0,1,1,1,0};
const int slsh[8] = {0,1,0,1,1,1,1};
const int col[8] = {0,1,1,1,0,1,0};
const int scol[8] = {0,1,1,1,0,1,1};
const int lthn[8] = {0,1,1,1,1,0,0};
const int equl[8] = {0,1,1,1,1,0,1};
const int gthn[8] = {0,1,1,1,1,1,0};
const int ques[8] = {0,1,1,1,1,1,1};
const int at[8] = {1,0,0,0,0,0,0};
const int lcbk[8] = {1,1,1,1,0,1,1};
const int rcbk[8] = {1,1,1,1,1,0,1};
const int til[8] = {1,1,1,1,1,1,0};
const int spac[8] = {0,1,0,0,0,0,0}; */

//*****receptor*****

#include <msp430.h>
#include <stdio.h>
#include <string.h>

// Variaveis
int adc[1] = {0};
int avg_adc = 0;

// Funcoes
void adc_Setup();
void adc_Sam10();

void main()

```

```

{
    WDTCTL = WDTPW + WDTHOLD;           // Stop WDT
    adc_Setup();                         // Chama adc_setup

    while(1)
    {
        adc_Sam10();                    // Chama adc_samp
    }
}

// ADC10 interrupt service routine
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    __bic_SR_register_on_exit(CPUOFF);   // Clear CPUOFF bit from 0(SR)
}

// funcao de setup
void adc_Setup()
{
    ADC10CTL1 = CONSEQ_2 + INCH_0;       // Repeat single channel, A0
    ADC10CTL0 = ADC10SHT_2 + MSC + ADC10ON + ADC10IE; // Sample & Hold Time +
ADC10 ON + Interrupt Enable
    ADC10DTC1 = 0x01;                   // define o numero de conversoes
    ADC10AE0 |= 0x01;                   // P1.0 ADC option select
}

// ADC sample conversion function
void adc_Sam10()
{
    ADC10CTL0 &= ~ENC;                   // Disable Conversion
    while (ADC10CTL1 & BUSY);            // Wait if ADC10 busy
    ADC10SA = (int)adc;                  // transfere informacao pro proximo vetor (DTC)
    ADC10CTL0 |= ENC + ADC10SC;         // Enable Conversion and conversion start
    __bis_SR_register(CPUOFF + GIE);    // Low Power Mode 0, ADC10_ISR
}

```

## 1.2 Códigos MATLAB

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Decoding.m%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Importar os dados do arquivo .dat criado pelo CCS
Dado = importdata('arquivoccs5', ' ', 1);
Inf=Dado.data(:, 1);

%Cria a string do header que será usado para comparar com a mensagem
headString = 'abc';
header = text2bin(headString);

%Determina o thresold tirando uma média de todos os dados
sum = 0;

```

```

for i=1:length(Inf)
    sum = sum + Inf(i);
end
threshold = sum/length(Inf);

%Converte os niveis de tensão de volta em 0s e 1s
for i=1:length(Inf)
    if(Inf(i) > threshold)

        Inf(i) = 1;
    else
        Inf(i) = 0;
    end
end
Inf=Inf';

% %Correlação e conversão da mensagem de volta em texto
corr = xcorr(Inf,header);
maxCorr = max(corr);
ind = find(corr >= maxCorr);
startData = ind - length(Inf) + length(header)+1 ;
msg = Inf(startData:startData +55) ' ; % 22+55=77 (tamanho da msg)
bin2text(msg')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%bin2text.m%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% transforma um vetor de 7 bits binario 0-1 em uma string
function ztext=bin2text(z)
rp=floor(length(z)/7);
rez=num2str(z(1:7*rp)')';
ztext=char(bin2dec(reshape(rez,7,rp)'))');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%text2bin.m%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% transforma uma string em um vetor binario 0-1
function n=text2bin(textstring)
bintext=dec2bin(double(textstring));
% texto em binario
[rp,cp]=size(bintext);
n=str2num(reshape(bintext',1,rp*cp)')');

```