

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**PROJETO E DESENVOLVIMENTO DE UM
SISTEMA DE GERENCIAMENTO REMOTO SEM
FIO APLICADO À ILUMINAÇÃO**

DISSERTAÇÃO DE MESTRADO

Diogo Ribeiro Vargas

Santa Maria, RS, Brasil

2012

**PROJETO E DESENVOLVIMENTO DE UM
SISTEMA DE GERENCIAMENTO REMOTO SEM
FIO APLICADO À ILUMINAÇÃO**

por

Diogo Ribeiro Vargas

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração em Processamento de Energia, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Engenharia Elétrica.**

Orientador: Prof. Dr. Alexandre Campos

Santa Maria, RS, Brasil

2012

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Vargas, Diogo Ribeiro

Projeto e desenvolvimento de um sistema de gerenciamento remoto sem fio aplicado à iluminação / Diogo Ribeiro Vargas. - 2012.

71 f. ; 30cm.

Orientador: Alexandre Campos

Dissertação (mestrado) – Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, RS, 2012.

1. Iluminação. 2. Multi-interfaces. 3. Gerenciamento Remoto. 4. Comunicação. 5. Sem-fio. I. Campos, Alexandre, orient. II. Título.

©2012

Todos os direitos autorais reservados a Diogo Ribeiro Vargas. A reprodução de partes ou do todo deste trabalho só poderá ser feita com autorização por escrito do autor.

Endereço: Av. Roraima, 1000. Prédio 10 (CT). Sala 530 (NUPEDEE/GEDRE), Bairro: Camobi, Santa Maria, RS, 97.105-900

Endereço Eletrônico: vargas@gedre.ufsm.br ou diogodevargas@gmail.com

**Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica**

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**PROJETO E DESENVOLVIMENTO DE UM SISTEMA DE
GERENCIAMENTO REMOTO SEM FIO APLICADO À
ILUMINAÇÃO**

elaborada por
Diogo Ribeiro Vargas

como requisito parcial para obtenção do grau de
Mestre em Engenharia Elétrica

COMISSÃO EXAMINADORA:

Alexandre Campos, Dr.
(Presidente/Orientador)

Gustavo Weber Denardin, Dr. (UTFPR)

Carlos Henrique Barriquello, Dr. (UFSM)

Santa Maria, 17 de Agosto de 2012

A minha esposa

AGRADECIMENTOS

Meus sinceros agradecimentos:

Primeiramente, gostaria de agradecer à minha família, por todo apoio e amor dedicados a mim. Em especial à minha esposa Michele, pela paciência e companheirismo.

Agradeço ao professor Alexandre Campos pela orientação, amizade e confiança no meu trabalho.

Aos professores do grupo GEDRE-Inteligência em Iluminação, Ricardo Nederson do Prado, Álysson Raniere Seidel, Marco Antônio Dalla Costa e Tiago Marchesan, não apenas pelo conhecimento técnico transmitido mas também por todo o apoio que dedicam ao grupo.

Aos colegas de Pós-Graduação André Luís Kirsten, Douglas Camponogara, Jacson Hansen, Juliano de Pelegrini Lopes, Maicol Melo, Marcelo Freitas da Silva, Marcelo Co-setin, Paulo Cesar Luz, Rafael Adaime Pinto, Victor Oberto e Vítor Bender.

Meu agradecimento em especial aos professores Carlos Henrique Barriquello e Gustavo Weber Denardin que aceitaram fazer parte da banca examinadora e por suas contribuições que muito contribuíram no desenvolvimento deste trabalho.

Estendo meu agradecimento a todos alunos de graduação do GEDRE, pelo seu auxílio aos trabalhos desenvolvidos pelo grupo. Principalmente aos graduandos Fernanda Mendes e William Vizzotto que me auxiliaram neste trabalho.

Agradeço também aos funcionários do NUPEDEE, Anacleto Brondani, Luiz Fernando Martins e Zulmar Nascimento, aos funcionários do Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), Carlo Castelanelli e Luciana Kapelinski, bem como, ao professor José Renes Pinheiro, coordenador do PPGEE.

E finalmente, gostaria de agradecer à comunidade de software livre por propiciar muitas das ferramentas utilizadas no desenvolvimento desta dissertação.

“If you have an apple and I have an apple and we exchange apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas.”

George Bernard Shaw

“Se você tem uma maçã e eu tenho uma maçã, e nós trocamos nossas maçãs, então você e eu ainda teremos apenas uma maçã cada. Mas se você tem uma idéia e eu tenho uma idéia, e nós trocamos essas idéias, então cada um de nós terá duas idéias.”

George Bernard Shaw

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Santa Maria, RS, Brasil

PROJETO E DESENVOLVIMENTO DE UM SISTEMA DE GERENCIAMENTO REMOTO SEM FIO APLICADO À ILUMINAÇÃO

AUTOR: DIOGO RIBEIRO VARGAS
ORIENTADOR: ALEXANDRE CAMPOS

Local da Defesa e Data: Santa Maria, 17 de Agosto de 2012.

Seja em parques, praças, jardins, centros de compras ou estacionamento, o campo de aplicação de sistemas de iluminação com fins decorativos, comerciais e artísticos, tem se expandido muito, principalmente, devido as fontes de luzes disponíveis atualmente. Este trabalho apresenta o desenvolvimento de um sistema de gerenciamento aplicado à iluminação baseado em LEDs RGB de potência, devido seu grande potencial para uso em iluminação decorativa de ambientes externos. O sistema resultante é uma solução de baixo custo que apresenta um grande diferencial, a possibilidade de gerenciamento remoto em múltiplas interfaces, seja por um dispositivo dedicado, quanto por dispositivos portáteis, como *notebooks*, *smartphones* ou *tablets*. Afim de se obter uma solução confiável e robusta, é desenvolvido também um protocolo de comunicação para o sistema proposto, após uma análise dos protocolos mais utilizados atualmente em sistemas de iluminação. Ao final deste trabalho são apresentados os resultados do sistema proposto, aonde também são levantadas as principais vantagens e limitações da solução apresentada.

Palavras-chave: Iluminação, Multi-interfaces, Gerenciamento remoto, Comunicação, Sem-fio

ABSTRACT

Master's Dissertation

Post Graduate Program in Electrical Engineering

Federal University of Santa Maria, RS, Brazil

PROJECT AND DEVELOPMENT OF A WIRELESS REMOTE MANAGEMENT SYSTEM APPLIED TO LIGHTING

AUTHOR: DIOGO RIBEIRO VARGAS

SUPERVISOR: ALEXANDRE CAMPOS

Place and Date: Santa Maria, August 17th, 2012.

Today's light sources are expanding the application field of lighting systems for decorative, commercial and artistic purposes, in many areas, such as parks, gardens or parking. This paper presents the development of a management system applied to lighting based on power RGB LEDs, due to its great potential for use in outdoor decorative lighting. The proposed system is a low cost solution and presents the advantage of remote management of the lighting system on multiple interfaces, either on a dedicated device, as for portable devices, such as laptops, smartphones or tablets. After an analysis of currently lighting systems protocols, it is also proposed a new communication protocol for the developed system in order to get a reliable and robust solution. This work also presents the results of the proposed system, highlighting its main advantages and limitations.

Keywords: Lighting, Multi-interfaces, Remote Management, Communications, Wireless, Decorative Lighting

SUMÁRIO

1	Introdução	p. 19
1.1	Visão geral	p. 19
1.2	Motivação	p. 20
1.3	Organização textual	p. 22
2	Sistemas de iluminação	p. 24
2.1	Iluminação	p. 24
2.2	Fontes de luz	p. 26
2.3	Diodos emissores de luz	p. 27
2.4	Sintetização de luz branca com LEDs	p. 28
3	Protocolos aplicados à iluminação	p. 31
3.1	Protocolo DALI TM	p. 31
3.2	Protocolo DMX512-A	p. 35
3.3	Análise comparativa	p. 36
4	Sistema proposto	p. 38
4.1	Introdução	p. 38
4.2	Circuito de acionamento dos LEDs	p. 39
4.3	Controlador do sistema	p. 42
4.3.1	Protocolo Desenvolvido	p. 44
4.4	Interface com dispositivos portáteis	p. 49
4.4.1	Hardware para comunicação Wi-Fi TM	p. 50

5 Resultados Experimentais	p. 56
5.1 Resultados para cor azul escuro	p. 56
5.2 Resultados para cor vermelho vinho	p. 57
5.3 Resultados para cor verde escuro	p. 59
6 Conclusão	p. 60
6.1 Sugestões de trabalhos futuros	p. 61
Referências	p. 63
Apêndice A – Código HTML implementado	p. 66

LISTA DE FIGURAS

FIGURA 1.1	Exemplo de iluminação decorativa de praças e jardins. Fonte (FAEBER, 2012)	p. 19
FIGURA 1.2	Iluminação a LED utilizada no monumento ao Cristo Redentor. Fonte (OSRAM, 2012)	p. 20
FIGURA 1.3	Projetores Traxon [®] utilizados na iluminação do Cristo Redentor. (a) Modelo Liner Shield AC XB 27 (b) Modelo Wall Washer Shield AC XB-36.	p. 21
FIGURA 1.4	Interface homem-máquina do controlador DMX512-A Leviton [®] Série 1000. Fonte (LEVITON, 2006)	p. 21
FIGURA 1.5	Diagrama de blocos geral de um sistema de iluminação com gerenciamento	p. 22
FIGURA 2.1	Sensibilidade do olho humano e de insetos para diversos comprimentos de onda. Adaptado de (BARGHINI; MEDEIRO, 2006)	p. 24
FIGURA 2.2	Sensibilidade relativa dos cones presentes no olho humano. Adaptado de (HAJIABOLI; POPOVIC, 2008)	p. 25
FIGURA 2.3	Resposta relativa do olho humano para visão escotópica e fotópica	p. 26
FIGURA 2.4	Fontes de luz. (a) Lâmpada Incandescente. (b) Lâmpada Fluorescente Compacta. (c) Lâmpada de Descarga em Alta Pressão. (d) Lâmpada Fluorescente Tubular. Adaptado de (OSRAM, 2012)	p. 27
FIGURA 2.5	Junção P-N em um diodo	p. 27
FIGURA 2.6	Polarização diodo e LED. (a) Polarização reversa. (b) Polarização direta diodo. (c) Polarização direta LED	p. 28
FIGURA 2.7	Luz emitida por LEDs de acordo com sua dopagem. Adaptado de (SCHUBERT, 2003)	p. 29
FIGURA 2.8	Espectro do fluxo radiante emitido por um LED <i>cool white</i>	p. 29
FIGURA 2.9	Espectro do fluxo radiante emitido por um LED RGB GM-P3WRGB-30-A para valores nominais de corrente	p. 30
FIGURA 3.1	Exemplo de codificação <i>Manchester</i> . Fonte (NEC, 2009a)	p. 32
FIGURA 3.2	Topologia Mestre-Escravo do padrão DALI TM	p. 32
FIGURA 3.3	Padrão de mensagem DALI TM (a) Comunicação Mestre-Escravo, (b) Comunicação Escravo-Mestre	p. 33
FIGURA 3.4	Relação logarítmica utilizada pelo padrão DALI TM na dimerização de lâmpadas	p. 34

FIGURA 3.5	Padrão de mensagem do protocolo DMX512-A	p. 35
FIGURA 3.6	Topologia mestre-escravo do protocolo DMX512-A	p. 36
FIGURA 4.1	Diagrama de blocos de um sistema de gerenciamento aplicado a sistemas de iluminação com múltiplas interfaces	p. 39
FIGURA 4.2	LED RGB de potência modelo GM-P3WRGB-30-A	p. 40
FIGURA 4.3	Circuito de <i>driver</i> para LEDs RGB de potência utilizado. Adaptado de (SÁ JUNIOR, 2010)	p. 41
FIGURA 4.4	Protótipo de <i>driver</i> e lâmpada desenvolvidos	p. 41
FIGURA 4.5	Diagrama de blocos de uma solução com controlador dedicado	p. 42
FIGURA 4.6	Rádios utilizados para comunicação RF433MHz, modelo TWS-DS-3 (transmissor) e RWS-375-6 (receptor)	p. 43
FIGURA 4.7	Plataforma de desenvolvimento Colduino	p. 44
FIGURA 4.8	Módulos de comunicação do protótipo desenvolvido. (a) Módulo Transmissor. (b) Módulo Receptor	p. 45
FIGURA 4.9	Padrão de mensagem do protocolo desenvolvido	p. 46
FIGURA 4.10	Comando para troca da cor RGB do protocolo desenvolvido	p. 47
FIGURA 4.11	Diagrama de blocos da solução com controlador dedicado apresentada	p. 49
FIGURA 4.12	Diagrama de blocos de uma solução com interface para dispositivos móveis	p. 50
FIGURA 4.13	Roteador Wi-Fi™ TP-Link® Modelo WR703N	p. 50
FIGURA 4.14	Terminal Telnet para configuração e customização do OpenWrt	p. 51
FIGURA 4.15	Porta RS232 do roteador Wi-Fi™ utilizado	p. 52
FIGURA 4.16	Interface de gerenciamento desenvolvida para dispositivos portáteis	p. 53
FIGURA 4.17	Diagrama de blocos da solução com interface para dispositivos móveis apresentada	p. 54
FIGURA 4.18	Módulo mestre com interface Wi-Fi™	p. 54
FIGURA 4.19	Módulo escravo - Receptor, driver e lâmpada RGB	p. 55
FIGURA 5.1	Cor azul escuro	p. 56
FIGURA 5.2	Cor azul escuro: Mensagem enviada pelo mestre (Canal 1 - $2V/div$) e recebida pelo escravo (Canal 3 - $2V/div$); Escala de tempo 17, 76ms	p. 57
FIGURA 5.3	Sinal PWM para dimerização da cor azul: Nó mestre (Canal 2 - $2V/div$) e nó escravo (Canal 4 - $2V/div$); Escala de tempo 17, 76ms	p. 57
FIGURA 5.4	Cor vermelho vinho	p. 58
FIGURA 5.5	Cor vinho: Mensagem enviada pelo mestre (Canal 1 - $2V/div$) e recebida pelo escravo (Canal 3 - $2V/div$); Escala de tempo 17, 76ms	p. 58

- FIGURA 5.6 Cor vinho: Sinal PWM no mestre (Canal 2 - $2V/div$) e escravo (Canal 4 - $2V/div$); Escala de tempo $17,76ms$ p. 58
- FIGURA 5.7 Cor verde escuro p. 59
- FIGURA 5.8 Cor verde escuro: Mensagem enviada pelo mestre (Canal 1 - $2V/div$) e recebida pelo escravo (Canal 3 - $2V/div$); Escala de tempo $17,76ms$ p. 59
- FIGURA 5.9 Cor verde escuro: dimerização no mestre (Canal 2 - $2V/div$) e no escravo (Canal 4 - $2V/div$); Escala de tempo $17,76ms$ p. 59

LISTA DE TABELAS

TABELA 3.1	Principais comandos do padrão DALI™	p. 33
TABELA 3.2	Memória não volátil utilizada pelo padrão DALI™. Adaptado de (TEXAS INSTRUMENTS, 2009)	p. 34
TABELA 3.3	Padrão de tempos do protocolo DM512-A	p. 36
TABELA 3.4	Comparativo entre os protocolos DALI™ e DMX512-A	p. 37
TABELA 4.1	Especificações do LED RGB de potência modelo GM-P3WRGB-30-A	p. 40
TABELA 4.2	Especificações dos rádios Wenshing TWS-DS-3 e Wenshing RWS-375-6	p. 43
TABELA 4.3	Especificações da plataforma de desenvolvimento Colduino	p. 44
TABELA 4.4	Comparativo entre o protocolo desenvolvido e os protocolos DALI™ e DMX512-A	p. 46
TABELA 4.5	Equivalência entre os comando do protocolo desenvolvido e comandos do padrão DALI™	p. 47
TABELA 4.6	Especificações do roteador Wi-Fi™ TP-Link® Modelo WR703N	p. 51

LISTA DE ALGORITMOS

- 4.1 Função para envio da cor RGB implementada no nó mestre p.48
- 4.2 Função para decodificação do protocolo implementada nos nós escravos . p.48
- 4.3 *Script* utilizado para envio de cores pela porta RS232 do roteador p.53

LISTA DE SÍMBOLOS

ACRÔNIMOS

ACN	<i>Architecture for Control Networks</i>
AM	<i>Amplitude Modulation</i>
ASK	<i>Amplitude-shift keying</i>
BACnet	<i>Building Automation and Control Networks</i>
bps	<i>Bits por segundo</i>
CRC	<i>Cyclic Redundancy Check</i>
DALI	<i>Digital Addressable Lighting Interface</i>
DMX512-A	<i>Digital Multiplex with 512 pieces of information</i>
HID	<i>High-intensity discharge</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEC	<i>International Electrotechnical Commission</i>
IHM	Interface Homem-Máquina
IP	<i>Internet Protocol</i>
IRC	<i>Índice de Reprodução de Cores</i>
ISO	<i>International Organization for Standardization</i>
KNX	<i>Konnex</i>
LED	<i>Light Emitting Diode</i>
LMS	<i>Long, Medium and Short</i>
PWM	<i>Pulse-width modulation</i>
RF	Rádio-Frequência
RGB	<i>Red, Green and Blue</i>
RX	Recepção
SPI	<i>Serial Peripheral Interface</i>

Lista de Símbolos

SSH	<i>Secure Shell</i>
TCP	<i>Transmission Control Protocol</i>
TX	Transmissão
USB	<i>Universal Serial Bus</i>

1 INTRODUÇÃO

Neste primeiro capítulo, apresenta-se uma visão geral do trabalho desenvolvido nesta dissertação.

1.1 Visão geral

Existem diversas aplicações para sistemas de iluminação artificial, seja ela para iluminação pública e de estacionamentos, iluminação de ambientes residenciais e industriais, iluminação de parques, praças e jardins, como também para iluminação decorativa, artística e comercial.

Desses campos de aplicação de sistema de iluminação, um que tem se expandido consideravelmente nos últimos anos é o de iluminação com fins decorativos e artísticos, como o apresentado na Fig.1.1.



Figura 1.1: Exemplo de iluminação decorativa de praças e jardins. Fonte (FAEBER, 2012)

Cabe destacar que um dos fatores que impulsionam essa expansão é a redução do custo de fontes de luz como os LEDs, que apresentam características de fácil alteração

na intensidade luminosa emitida (dimerização) e possibilidade de sintetização de diversas cores com o uso de diferentes LEDs.

Pode-se usar como um exemplo de iluminação decorativa baseada em LEDs o sistema de iluminação do monumento ao Cristo Redentor na cidade do Rio de Janeiro. O sistema foi implantado através de uma parceria entre a prefeitura da cidade, a arquidiocese e a empresa OSRAM. A iluminação deste cartão postal da cidade do Rio de Janeiro conta com 300 projetores de LEDs e um sistema de gerenciamento que permite alterar, individualmente, a cor, a intensidade e o direcionamento de cada um dos projetores (OSRAM, 2012). O monumento iluminado está ilustrado pela Fig.1.2.



Figura 1.2: Iluminação a LED utilizada no monumento ao Cristo Redentor. Fonte (OSRAM, 2012)

Os projetores de LEDs utilizados são mostrados na Fig.1.3. Esses projetores são fabricados pela empresa Traxon Technologies[®] (empresa do grupo OSRAM), sendo 40 projetores do modelo Liner Shield AC XB 27 (TRAXON, 2010a) e 260 do modelo Wall Washer Shield AC XB-3 (TRAXON, 2010b).

1.2 Motivação

Aplicações como iluminação decorativa, teatral e artística podem necessitar de uma dinâmica para alterar não apenas a intensidade luminosa, mas também a cor da luz sintetizada pelo sistema de iluminação, o que acarreta em uma maior complexidade no gerenciamento do sistema e conseqüentemente um maior custo.

Atualmente existem diversos padrões de protocolo de comunicação utilizados para controle de iluminação. Por exemplo, o protocolo mais utilizado em componentes profis-



Figura 1.3: Projetores Traxon[®] utilizados na iluminação do Cristo Redentor. (a) Modelo Liner Shield AC XB 27 (b) Modelo Wall Washer Shield AC XB-36.

sionais de iluminação de teatros é o padrão DMX512-A (PANDYA, 2007), (NEC, 2009b). Controladores desse tipo de sistemas, normalmente, apresentam uma interface complexa. Por exemplo, o controlador Leviton[®] Série 1000 que é considerado um controlador compacto e simples para sistemas DMX512-A, tem sua interface mostrada na Fig.1.4.



Figura 1.4: Interface homem-máquina do controlador DMX512-A Leviton[®] Série 1000. Fonte (LEVITON, 2006)

O projeto de um sistema de gerenciamento remoto de baixo custo apresenta diversos desafios. Podendo-se destacar aqui, o desenvolvimento da Interface Homem-Máquina (IHM) entre o usuário e o controlador. Outro desafio é o desenvolvimento da comunicação entre o controlador e o circuito de acionamento (*driver*) da lâmpada. A Fig.1.5 apresenta em um diagrama de blocos as principais atribuições de um sistema de iluminação com

gerenciamento remoto.

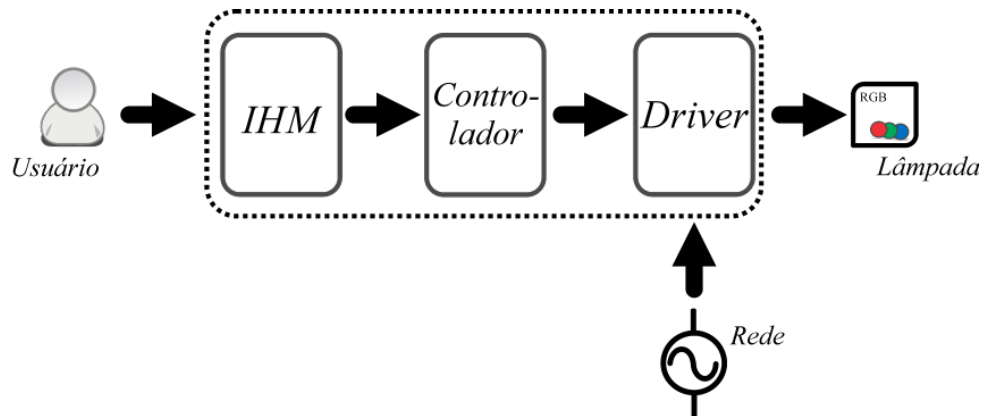


Figura 1.5: Diagrama de blocos geral de um sistema de iluminação com gerenciamento

O principal objetivo deste trabalho é o desenvolvimento de uma solução de baixo custo para o gerenciamento de sistemas de iluminação decorativa, com aplicação em ambientes externos (*outdoor*) e que apresente possibilidade de gerenciamento remoto. A solução ainda deve apresentar características como facilidade de instalação e de utilização por parte do usuário (*friendly use*). Sendo que o alcance desejado, para o gerenciamento remoto, é de uma distância de no mínimo 35 metros.

Assim, nesta dissertação, foi desenvolvida uma solução para esse objetivo, que apresenta um grande diferencial, a possibilidade de se utilizar múltiplas interfaces para o gerenciamento do sistema de iluminação. Pois é possível realizar as tarefas de gerenciamento por meio de diferentes dispositivos, seja por um dispositivo dedicado, ou pela utilização de dispositivos portáteis, como *notebooks*, *smartphones* ou *tablets*.

Para obter uma solução confiável e robusta, foi desenvolvido também um protocolo de comunicação para o sistema, após uma análise dos protocolos mais utilizados atualmente em sistemas de iluminação.

1.3 Organização textual

Este trabalho está dividido em seis capítulos. Sendo apresentada uma introdução aos sistemas de iluminação artificial no capítulo 2, em que também é analisada a aplicação de LEDs em sistemas de iluminação decorativas.

No capítulo 3 são analisados os principais protocolos de comunicação utilizados atualmente para gerenciamento de sistemas de iluminação.

A solução proposta é apresentada e desenvolvida no capítulo 4. E os resultados,

características e limitações dessa solução são mostrados no capítulo 5. Por fim, o capítulo 6 apresenta a conclusão deste trabalho.

2 SISTEMAS DE ILUMINAÇÃO

Neste capítulo, é apresentada uma breve introdução aos sistemas de iluminação artificial, com foco nos LEDs, por serem a fonte de luz que melhor atendem aos requisitos deste trabalho.

2.1 Iluminação

A iluminação tem um grande impacto sobre a vida na terra, sendo que há pelo menos três bilhões de anos os seres vivos apresentam relógios biológicos desenvolvidos, a fim de se adaptar às alternâncias naturais de dias e noites (luz e obscuridade) (BARGHINI; MEDEIRO, 2006).

Durante a evolução, os seres vivos desenvolveram diferentes formas de perceber as ondas eletromagnéticas emitidas pelo sol, por exemplo, enquanto os insetos percebem luz de aproximadamente 280 até 680nm, a faixa que os seres humanos percebem está aproximadamente entre 380 e 780nm. Para fins de comparação, as sensibilidades do sistema de visão destes animais são mostradas na Fig. 2.1.

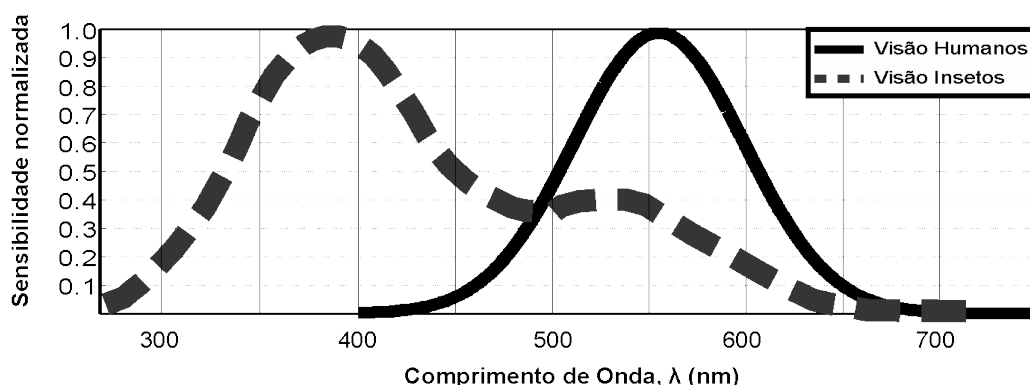


Figura 2.1: Sensibilidade do olho humano e de insetos para diversos comprimentos de onda. Adaptado de (BARGHINI; MEDEIRO, 2006)

No olho humano existem dois tipos de células responsáveis pela recepção de luz, são elas os cones e os bastonetes. Sendo os bastonetes responsáveis por uma visão monocro-

mática, identificando a intensidade da luz, e os cones responsáveis por perceberem três cores básicas: o vermelho, o verde e o azul (*Red, Green and Blue, RGB*).

Os cones podem ser divididos em três tipos de fotorreceptores: L, M e S (*Long, Medium and Short*), responsáveis por perceber três comprimentos de ondas diferentes (STOCKMAN; SHARPE; FACH, 1999), (DE PAULA, 2006), (HAJIABOLI; POPOVIC, 2008).

Cones L (*Long*) são os responsáveis pela percepção de ondas eletromagnéticas com maior comprimento de onda e apresentam o pico da sensibilidade em aproximadamente 555nm (cor vermelha). Os cones M (*Medium*) tem seu pico de sensibilidade em aproximadamente 525nm (cor verde). E os cones S (*Short*) apresentam maior sensibilidade para ondas de 420nm (cor azul) (STOCKMAN; SHARPE, 2000).

A relação de sensibilidade de cada tipo de cones presentes no olho humano é mostrada na Fig. 2.2.

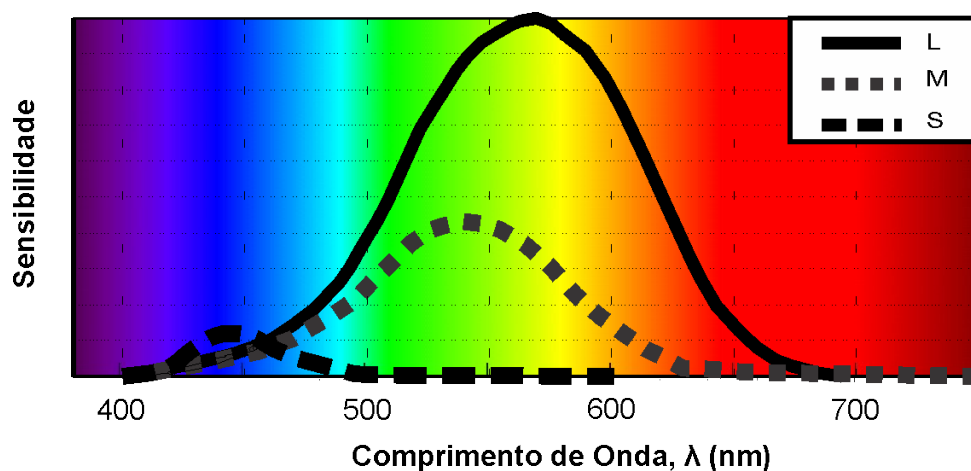


Figura 2.2: Sensibilidade relativa dos cones presentes no olho humano. Adaptado de (HAJIABOLI; POPOVIC, 2008)

Um fator que altera a resposta do sistema de visão humano é a densidade de luz a que o olho está submetido. Assim, quando o olho é submetido a uma luminância superior a $3,4\text{cd}/\text{m}^2$ (condição de visão diurna) os cones são os grandes responsáveis pela chamada visão fotópica, onde esses reagem rapidamente para detectar detalhes e cores. Porém em uma condição de visão noturna (luminância inferior a $0,034\text{cd}/\text{m}^2$) os receptores responsáveis pela visão são os bastonetes, e essa visão é chamada de visão escotópica (FRAYTAG et al., 2010).

Assim, devido à diferença de sensibilidade entre as células receptoras envolvidas em cada uma das visões, fotópica e escotópica, a sensibilidade resultante do olho humano também se modifica. A resposta do olho humano é demonstrada na Fig. 2.3 para cada uma das visões apresentadas.

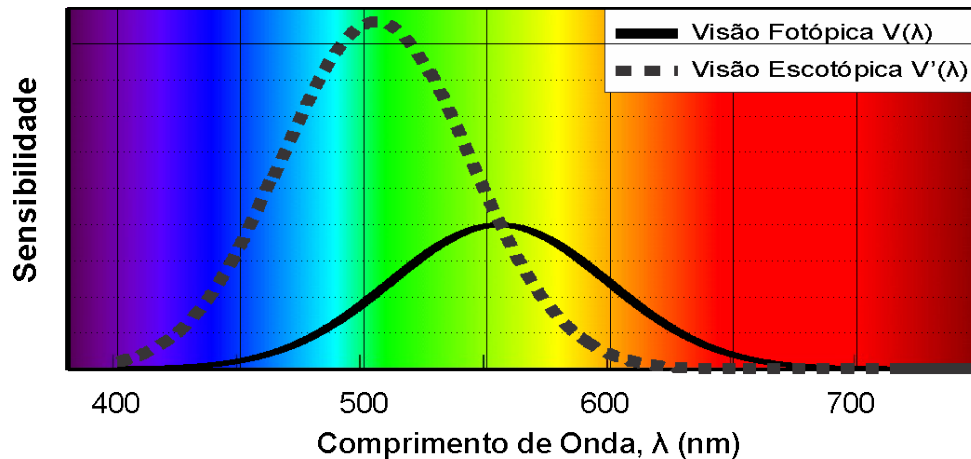


Figura 2.3: Resposta relativa do olho humano para visão escotópica e fotópica

2.2 Fontes de luz

Ainda na Era Pré-Histórica, os seres humanos dominaram a técnica do fogo. Este processo de geração de iluminação artificial evoluiu, passando por lampiões em 1874, pela lâmpada incandescente de Thomas Edison em 1879, pela lâmpada incandescente de filamento de tungstênio em 1993 e continua evoluindo.

As principais fontes de luz disponíveis atualmente podem ser classificadas em:

- Lâmpadas incandescentes;
- Lâmpadas de descarga:
 - Alta Pressão (*High-intensity discharge*, HID);
 - Baixa Pressão (Lâmpadas fluorescentes tubulares e compactas);
- Iluminação Semicondutora:
 - LEDs indicativos;
 - LEDs de alto brilho;
 - LEDs de potência.

Sistemas de iluminação baseados em Diodos Emissores de Luz (*Light Emitting Diode*, LED) são conhecidos por iluminação semicondutora ou iluminação de estado sólido. Esses dispositivos são uma ótima alternativa para iluminação de ambientes residenciais internos (*indoor*), devido às suas características como alta eficácia luminosa (relação lúmen por Watt, lm/W) e alto Índice de Reprodução de Cores (IRC).

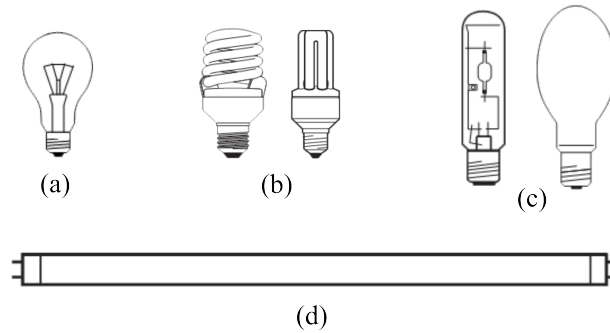


Figura 2.4: Fontes de luz. (a) Lâmpada Incandescente. (b) Lâmpada Fluorescente Compacta. (c) Lâmpada de Descarga em Alta Pressão. (d) Lâmpada Fluorescente Tubular. Adaptado de (OSRAM, 2012)

Outras características dos LEDs como o seu baixo ângulo de abertura e tamanho físico reduzido podem ser aproveitadas no projeto de luminárias. Assim, é possível obter um sistema de iluminação que pode ser facilmente ocultado no ambiente e proporcionar maior liberdade ao projeto arquitetônico.

Os LEDs ainda apresentam uma baixa inércia na ignição, facilidade na redução da intensidade luminosa (dimerização), capacidade de escolha de cor emitida e vida útil elevada (PINTO, 2009).

2.3 Diodos emissores de luz

Os diodos são dispositivos semicondutores formados por uma junção P-N, onde o lado N apresenta cargas negativas (excesso de elétrons) e o lado P apresenta cargas positivas (falta de elétrons). Assim, existem duas possibilidades de conexão entre diodos e fontes, chamadas de polarização direta e polarização reversa.

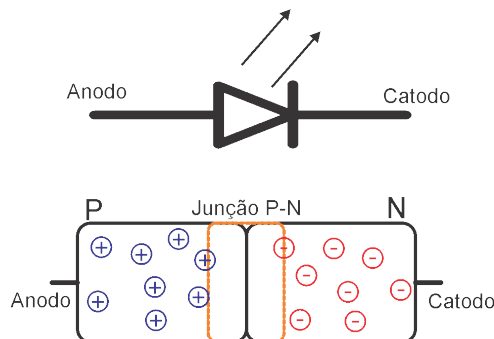


Figura 2.5: Junção P-N em um diodo

Quando um diodo é polarizado reversamente, acontece um rearranjo das cargas positivas e negativas, formando assim uma região de vazão de cargas na junção P-N, o que acaba por bloquear a circulação de cargas no dispositivo.

Entretanto, ao ser polarizado diretamente, as cargas do diodo se posicionam de maneira a formar um caminho que permite a circulação de corrente elétrica (elétrons). Em um diodo emissor de luz (LED), enquanto polarizado diretamente, parte da energia dos elétrons é liberada na forma de luz (fótons) no fenômeno chamado eletroluminescência (BULLOUGH, 2003).

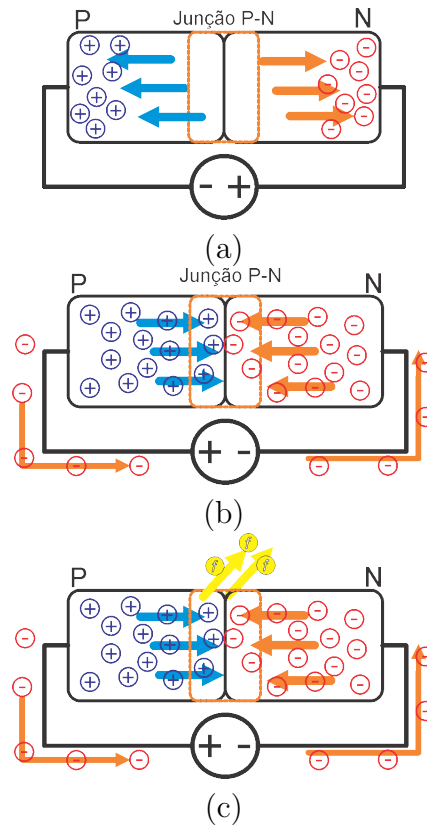


Figura 2.6: Polarização diodo e LED. (a) Polarização reversa. (b) Polarização direta diodo. (c) Polarização direta LED

A luz emitida por um LED depende dos materiais utilizados na dopagem das regiões P e N. Ao analisar o espectro eletromagnético emitido por dispositivos baseados em GaInN/GaN azul, é possível notar que o pico do comprimento de onda está em $470nm$. Já para dispositivos baseados em GaInN/GaN verde, este pico fica em $525nm$, e em LED de AlGaInP/GaAs vermelho em $625nm$ (SCHUBERT, 2003). A Fig. 2.7 mostra o espectro emitido por LEDs com diferentes dopagens.

2.4 Sintetização de luz branca com LEDs

A sintetização de luz branca a partir de LEDs normalmente é feita por duas técnicas, conversão da luz azul ou combinação de cores primárias.

Em um LED que emite luz com baixo comprimento de onda adiciona-se uma camada

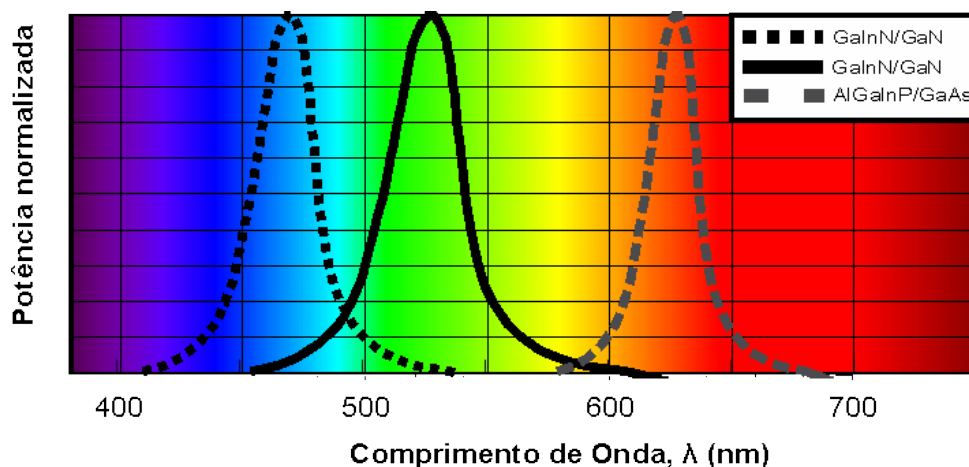


Figura 2.7: Luz emitida por LEDs de acordo com sua dopagem. Adaptado de (SCHUBERT, 2003)

de fósforo sobre o LED, assim parte desta luz é convertida para um comprimento de onda maior. Por exemplo um LED de luz azul tem o pico do comprimento de onda entre 450 e 470nm , ao adicionar uma camada de fósforo, parte desta luz é convertida para um comprimento de onda aproximadamente entre 500nm e 650nm (com um pico aproximadamente em 580nm), assim com a mistura destas luzes pode-se sintetizar uma luz branca (Bullough, 2003). A Fig.2.8 mostra o espectro do fluxo radiante emitido por um LED Philips Modelo LXML-PWC1-0100 (PHILIPS, 2011) que utiliza esta técnica para sintetizar luz branca com temperatura de cor de aproximadamente 6500K (*Cool-White*).

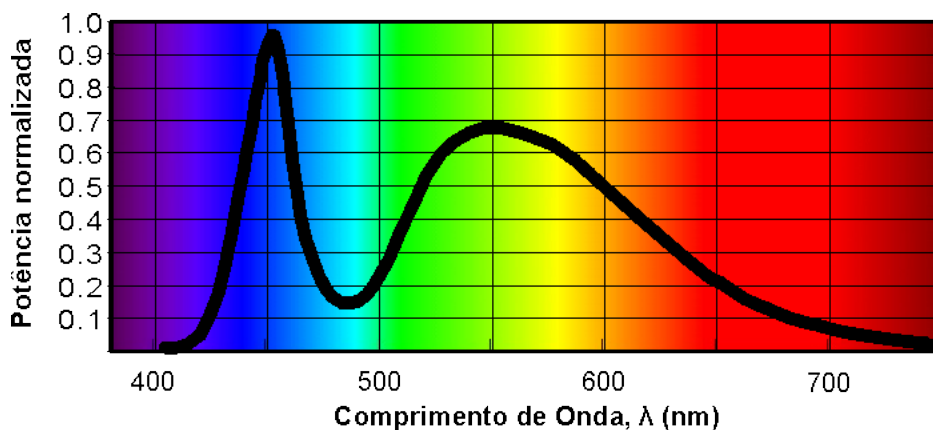


Figura 2.8: Espectro do fluxo radiante emitido por um LED *cool white*

Como visto na Seção 2.1 o olho humano apresenta três tipos de cones, assim o olho tem uma sensibilidade maior na percepção de três cores: vermelho, verde e azul. Devido a esta característica do olho humano, outra técnica que pode ser usada para sintetizar a luz branca é utilizar a correta mistura de três fontes de luz primárias, ou seja, utilizar três LEDs de cores vermelho ($625 - 740\text{nm}$), verde ($500 - 565\text{nm}$) e azul ($440 - 485\text{nm}$) (LEDs RGB).

O espectro emitido por um LED RGB de potência modelo GM-P3WRGB-30-A (GUANGMAI, 2009) é mostrado na Fig.2.9, para valores nominais de corrente e apresenta uma temperatura de cor de $6700K$. Este modelo de LED apresenta em único encapsulamento três LEDs, sendo o fluxo luminoso máximo emitido para a cor vermelha de 35 lm, 55 lm na cor verde e 15 lm para a cor azul.

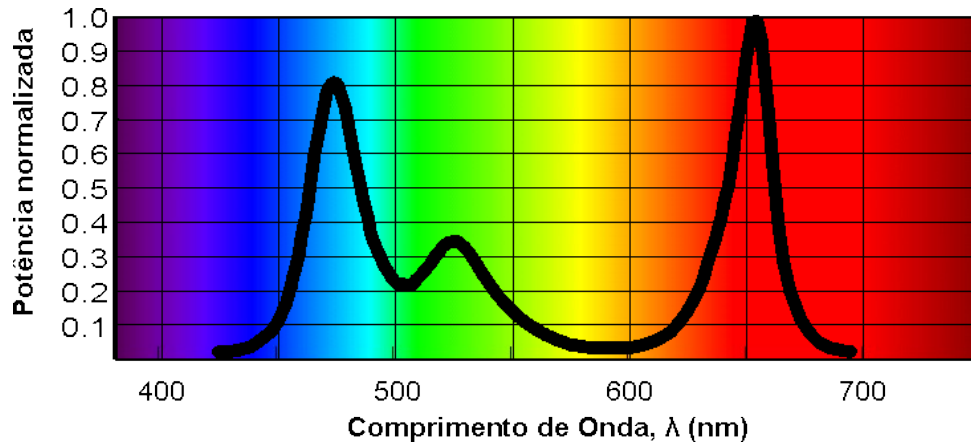


Figura 2.9: Espectro do fluxo radiante emitido por um LED RGB GM-P3WRGB-30-A para valores nominais de corrente

Tanto a Fig.2.8 quanto a Fig.2.9 foram obtidas através de experimentos em uma espera integradora e representam medidas radiométricas. Essas figuras demonstram a potência emitida pelos LEDs e estão normalizadas pelo valor máximo de potência (Watts).

Nota-se que o valor máximo da Fig.2.9 está em aproximadamente $650nm$ (cor vermelha), porém segundo as especificações do fabricante esse LED tem a capacidade de sintetizar mais fluxo luminoso na cor verde (55 lúmens entre $500 - 565nm$). Essa diferença deve-se ao fato de o fluxo luminoso ser uma unidade fotométrica, ou seja, leva em conta a percepção do olho humano e a Fig.2.9 representa uma medida radiométrica (fluxo radiante).

3 PROTOCOLOS APLICADOS À ILUMINAÇÃO

Atualmente existem diversos padrões de protocolos de comunicação utilizados para controle de iluminação, como por exemplo o padrão ISO 16484-5 (*Building Automation and Control Networks*, BACnet[®]), o padrão IEC 60929-E (*Digital Addressable Lighting Interface*, DALI[™]), o padrão ANSI E1.11 (*Digital Multiplex with 512 pieces of information*, DMX512-A), o padrão ANSI E1.17 (*Architecture for Control Networks*, ACN) e o padrão ISO/IEC 14543-3 (*Konnex*, KNX).

Devido sua ampla utilização na indústria (IES, 2011), os protocolos de rede DALI[™] e DMX512-A serão analisados mais detalhadamente neste capítulo.

3.1 Protocolo DALI[™]

O protocolo de comunicação DALI[™] é um protocolo aberto que foi criado inicialmente para ser utilizado apenas em aplicações com iluminação baseada em lâmpadas fluorescentes, sendo padronizado pela norma IEC 60929-E. Posteriormente este padrão foi atualizado pelas normas IEC 62386-101 e IEC 62386-102 para, entre outras coisas, incluir o suporte à iluminação com LEDs.

Atualmente encontram-se no mercado dispositivos DALI[™] para uso com diversas fontes de luz, como lâmpadas incandescentes, lâmpadas fluorescentes, lâmpadas de descarga em alta pressão (*High-intensity discharge*, HID) e LEDs (TRIDONIC, 2011).

Este padrão é mantido pelo *DALI Activity Group* (DALI AG) que está vinculada à Associação Alemã da Indústria Elétrica e Eletrônica (*Zentralverband Elektrotechnik- und Elektronikindustrie*, ZVEI) e seu uso é amplamente difundido na indústria (HEIN, 2001), (FREESCALE, 2002), (MICROCHIP, 2002), (RONAT, 2002), (CONTENTI, 2002), (ZHANG; ZHOU; WU, 2006), (NEC, 2009a), (TEXAS INSTRUMENTS, 2009), (TRIDONIC, 2011).

Para realizar a troca de dados, este protocolo utiliza uma comunicação serial assíncrona, *half-duplex*, com taxa de transferência de dados de 1200 bit/s.

Por padrão a codificação *Manchester* é utilizada em sua camada física, porém já existem aplicações que utilizam outros padrões como camadas físicas, por exemplo *Ethernet* (IES, 2011). Um exemplo de codificação *Manchester* é a Fig. 3.1.

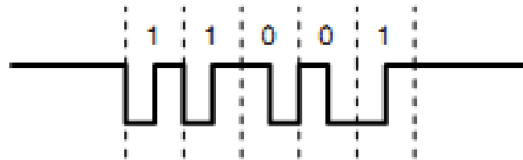


Figura 3.1: Exemplo de codificação *Manchester*. Fonte (NEC, 2009a)

Redes DALI™ são baseadas na topologia mestre-escravo, onde o usuário, através do dispositivo mestre (ou controlador), envia mensagens aos dispositivos escravos (no caso, reatores eletrônicos), podendo estas mensagens serem endereçadas a um escravo específico ou a um grupo de escravos. Para isso, cada escravo tem um endereço único (um valor entre 0 e 63), assim cada mestre pode controlar até 64 escravos diferentes.

Por apresentar uma comunicação bidirecional o mestre pode, através de comando, solicitar informações a um escravo específico. Porém, o início da comunicação sempre parte do mestre, ou seja, um escravo não pode enviar dados sem que lhe tenha sido requisitado.

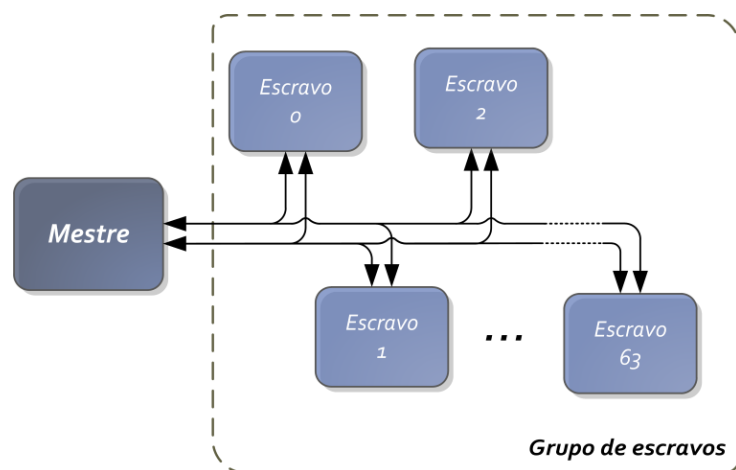


Figura 3.2: Topologia Mestre-Escravo do padrão DALI™

Existem diferenças entre as mensagens de comando (que são enviadas pelo mestre) e a resposta enviada pelos escravos.

Mensagens enviadas pelo mestre aos escravos seguem o padrão de um *start bit*, um *byte* de endereço, um *byte* de dados e dois *stop bits*, conforme a Fig. 3.3a. Porém, quando

um escravo responde ao seu mestre, segue outro padrão, 1 *start bit*, 1 *byte* de dados e dois *stop bits*, como visto na Fig. 3.3b.

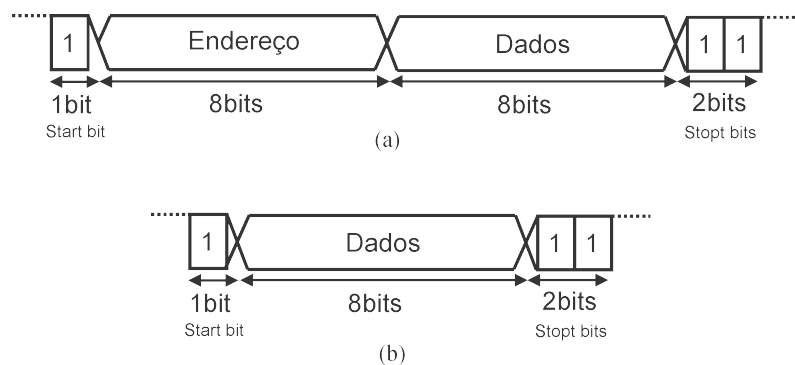


Figura 3.3: Padrão de mensagem DALI™ (a) Comunicação Mestre-Escravo, (b) Comunicação Escravo-Mestre

Este protocolo prevê 255 comandos que um mestre pode enviar aos seus escravos. Dentre esses, encontram-se comandos de ligar e desligar a lâmpada instantaneamente, comandos para escolha da potência da lâmpada (dimerização) e comandos de ligar e desligar suavemente, onde a potência da lâmpada é incrementada (ou decrementada) gradativamente até chegar no valor desejado. Além disso, o tempo que o incremento (ou decremento) deve durar também pode ser configurado por um comando específico. Um resumo com os principais comandos deste padrão é apresentado na Tabela 3.1.

Tabela 3.1: Principais comandos do padrão DALI™

	Comando	Descrição
<i>Off</i>	0x00	Desliga instantaneamente
<i>Up</i>	0x01	Aumenta o valor da potência por 200ms
<i>Down</i>	0x02	Reduz o valor da potência por 200ms
<i>Step Up</i>	0x03	Incrementa o valor de potência em uma unidade
<i>On and Step Up</i>	0x08	Se a lâmpada está desligada, liga no valor mínimo. Se já está ligada, incrementa a potência em uma unidade
<i>Step Down</i>	0x04	Decrementa o valor de potência em uma unidade. Quando alcança o valor mínimo, mantém este.
<i>Step Down and Off</i>	0x07	Decrementa o valor de potência em uma unidade. Se chegar no valor mínimo, desliga.
<i>Recall MIN Level</i>	0x06	Escolhe o valor de potência no mínimo
<i>Recall MAX Level</i>	0x05	Escolhe o valor de potência no máximo

A dimerização de uma lâmpada no padrão DALI™, não é feita de maneira linear e sim através de uma relação logarítmica entre o valor enviado pelo mestre (valor entre 0 e 255) e a potência entregue à lâmpada pelo reator eletrônico escravo.

Assim, tem-se uma precisão maior para pequenos valores de potência e uma precisão menor para os valores mais próximos da potência nominal, o que corresponde melhor ao comportamento do olho humano (IES, 2011). A Fig.3.4 mostra esta relação logarítmica.

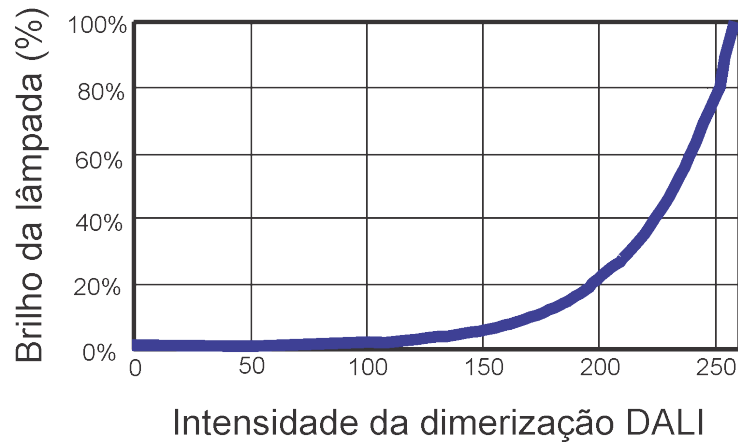


Figura 3.4: Relação logarítmica utilizada pelo padrão DALITM na dimerização de lâmpadas

Neste padrão todo escravo necessita de uma memória não volátil, para armazenar informações, como o seu endereço, o grupo ao qual está associado, e o tempo de incremento para o acionamento suave. Este custo de memória é mostrado na Tabela 3.2.

Tabela 3.2: Memória não volátil utilizada pelo padrão DALITM. Adaptado de (TEXAS INSTRUMENTS, 2009)

Variável	Tamanho [byte]
<i>Power On Level</i>	1
<i>System Failure Level</i>	1
<i>Minimum Level</i>	1
<i>Maximum Level</i>	1
<i>Fade Rate</i>	1
<i>Fade Time</i>	1
<i>Short Address</i>	1
<i>Group 0 through 7</i>	1
<i>Group 8 through 15</i>	1
<i>Scene 0 through 15</i>	15
<i>Random Address</i>	2
<i>Fast Fade Time</i>	1
<i>Failure Status</i>	1
<i>Operating Mode</i>	1
<i>Dimming Curve</i>	1
Total	30

3.2 Protocolo DMX512-A

O padrão DMX512-A está definido pela norma ANSI E1.11 e é desenvolvido pela Associação Americana de Serviços de Entretenimento e Tecnologia (*Entertainment Services and Technology Association*, ESTA). Atualmente é protocolo de comunicação usado na maioria dos componentes profissionais de iluminação de teatros (PANDYA, 2007), (NEC, 2009b).

Assim como o protocolo visto na seção 3.1, este padrão também utiliza topologia mestre-escravo e é baseado em uma comunicação serial. Onde o DMX512-A utiliza o protocolo RS485 (serial diferencial) e uma taxa de dados de 250k bit/s. Porém ao contrário do DALITM, o DMX512-A é um protocolo unidirecional. Ou seja, apenas o mestre pode enviar comandos aos escravos, e os escravos não podem retornar nenhum tipo de informação ao seu mestre.

Esse protocolo possui 512 canais para envio de mensagens. Sendo que os pacotes enviados pelo mestre contém os comandos de todos os 512 canais, ou seja, são enviados os 512 comandos toda vez que há o envio de mensagens.

Cada comando DMX512-A é composto de 8 *bits*. A estrutura padrão de mensagens desse protocolo é mostrada na Fig.3.5.

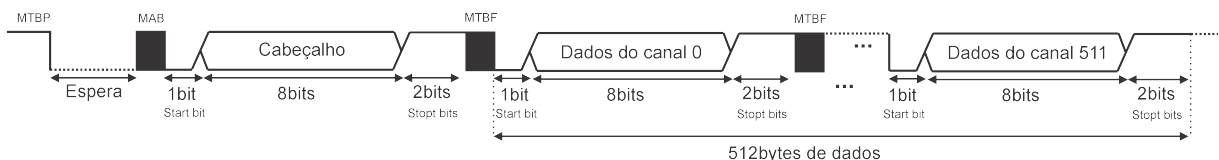


Figura 3.5: Padrão de mensagem do protocolo DMX512-A

O protocolo especifica quatro marcas, ou paradas, que devem ser respeitadas durante o envio de um pacote. Entre cada pacote existe um tempo (*Mark Time Between Packets*, MTBP) em que a linha de transmissão precisa ficar ociosa, ou seja, em nível alto. Após o MTBP, a linha fica em espera (nível lógico baixo) por uma marca (*Mark after Break*, MAB), que normalmente consiste em manter a linha em nível alto por $8\mu s$. Entre os dados de cada canal (*Mark Time Between Frames*, MTBF) o sistema fica esperando por um *start bit* por no máximo 1s.

Como se trata de um protocolo unidirecional, não existe a possibilidade dos escravos requisitarem a retransmissão de pacotes perdidos. Além disso, este protocolo não apresenta nenhum tipo de teste nos dados recebidos a fim de evitar erros e mensagens corrompidas. Com o intuito de evitar que algum dos escravos execute um comando incor-

Tabela 3.3: Padrão de tempos do protocolo DMX512-A

Nome	Tempo Mínimo	Tempo Típico	Tempo Máximo	Unidade
MTBP	0	-	1.000.000	μs
Espera	88	88	1.000.000	μs
MAB	8	8	1.000.000	μs
MTBF	0	-	1.000.000	μs

reto, normalmente, os mestres DMX512-A enviam os pacotes continuamente, mesmo que não haja mudança nos valores dos canais de dados.

Quando utiliza-se o padrão DMX512-A na dimerização de lâmpadas RGB é usual alocar três canais de dados para cada lâmpada, um para cada cor, a fim de se obter uma boa resolução na dimerização. Porém, isso acaba também diminuindo o número de escravos permitidos, limitando assim o uso de no máximo 170 lâmpadas RGB. A Fig.3.6 demonstra uma rede DMX512-A, onde os canais 0, 1 e 2 foram alocados para uma única lâmpada RGB.

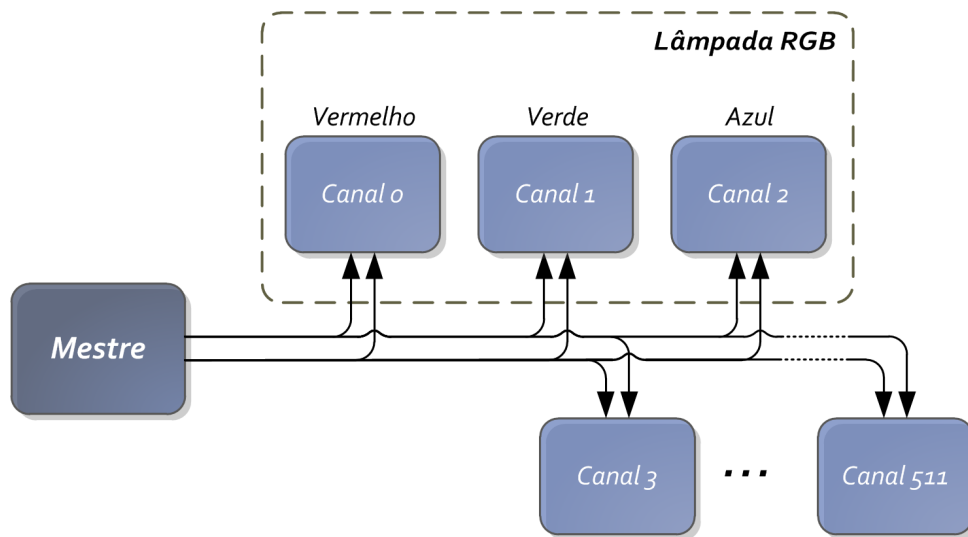


Figura 3.6: Topologia mestre-escravo do protocolo DMX512-A

3.3 Análise comparativa

Como visto anteriormente, ambos os protocolos (DALITM e DMX512-A) são amplamente utilizados na indústria em sistemas de iluminação, porém estes protocolos apresentam algumas limitações.

Apesar de sistemas baseados no padrão DALITM apresentarem a necessidade da presença de uma memória não volátil, isso não pode ser considerado um fator crítico, visto

que são necessários apenas *30bytes*, valor que pode ser encontrado na maioria dos microcontroladores de baixo custo disponíveis atualmente.

Além disso, a rede formada entre estes dispositivos é bidirecional, o que aumenta os custos em aplicações sem fio, devido a necessidade de utilizar-se rádios com capacidade de transmissão e recepção de dados em todos os dispositivos da rede.

Tabela 3.4: Comparativo entre os protocolos DALITM e DMX512-A

	DALITM	DMX512-a
Taxa de dados	1200 <i>bit/s</i>	250 <i>kbit/s</i>
Comandos	255	255
Escravos	64	512
Comunicação	Bidirecional	Unidirecional

Em relação ao DMX512-A, pode-se dizer que os mestres ocupam o canal de transmissão para o envio de pacotes durante todo tempo em que estiverem ligados. Pois como não existem testes dos dados recebidos pelos escravos, os mestres enviam continuamente os pacotes, mesmo que não existam alterações em nenhum dos comandos, a fim de diminuir o impacto do eventual recebimento de um pacote corrompido.

Porém, mesmo que esta retransmissão contínua fosse evitada pelo mestre, ainda haveria uma grande utilização do canal de comunicação, devido ao fato de o mestre necessitar enviar os dados de todos os seus 512 canais, mesmo que apenas um deles tenha sido alterado, ou ainda, mesmo em aplicações com poucos escravos é necessário enviar os dados de todos os 512 canais.

4 *SISTEMA PROPOSTO*

Este capítulo da dissertação engloba tanto o hardware quanto o software, desenvolvidos para solução do problema inicial.

4.1 Introdução

Como visto na Seção 1.2 o objetivo principal deste trabalho é obter um sistema de gerenciamento para iluminação decorativa de ambientes externos, onde os requisitos principais são:

- Utilização em iluminação decorativa;
- Baixo custo;
- Fácil utilização;
- Utilização em ambientes externos;
- Gerenciamento a uma distância de pelo menos 35 metros;
- Fácil instalação do sistema.

A fim de organizar e facilitar o desenvolvimento da solução, o objetivo principal foi dividido em objetivos parciais, baseados nos requisitos iniciais, de maneira que cada objetivo parcial é uma etapa do projeto.

Assim, ao se obter essas partes menores e integrá-las, obtêm-se a solução completa para o problema. Cada etapa passou por um fluxo de projeto que consistiu no desenvolvimento, teste, integração com o sistema e teste global.

Primeiramente, foi selecionado um *driver* para o grupo de LEDs de potência que será utilizado na lâmpada. Este circuito tem que ser capaz de manter a corrente nos LEDs

dentro dos níveis recomendados pelo fabricante e ainda, deve apresentar capacidade de dimerização dos LEDs.

Como é necessário variar a intensidade luminosa, de maneira independente, em cada uma das cores dos LEDs RGB, torna-se necessário que a utilização de um *driver* com capacidade de dimerização em cada uma das suas três saídas de forma independente.

A segunda etapa consiste no desenvolvimento de software e hardware utilizados para formar a comunicação entre o usuário e o *driver* projetado na primeira etapa. Ainda nesta etapa é exposto o protocolo de comunicação desenvolvido para ser utilizado neste sistema.

A terceira etapa consiste no desenvolvimento da interface homem-máquina, destaca-se o desenvolvimento de dois tipos de interface que permitem ao usuário o gerenciamento de todo o sistema de uma maneira fácil e intuitiva. Sendo que o primeiro tipo de interface desenvolvida é um controlador dedicado. E a segunda interface é tal que permite a utilização de diversos dispositivos portáteis. Assim, o sistema resultante é uma solução que permite o gerenciamento remoto do sistema de iluminação através de múltiplas interfaces, conforme ilustrado na Fig.4.1.

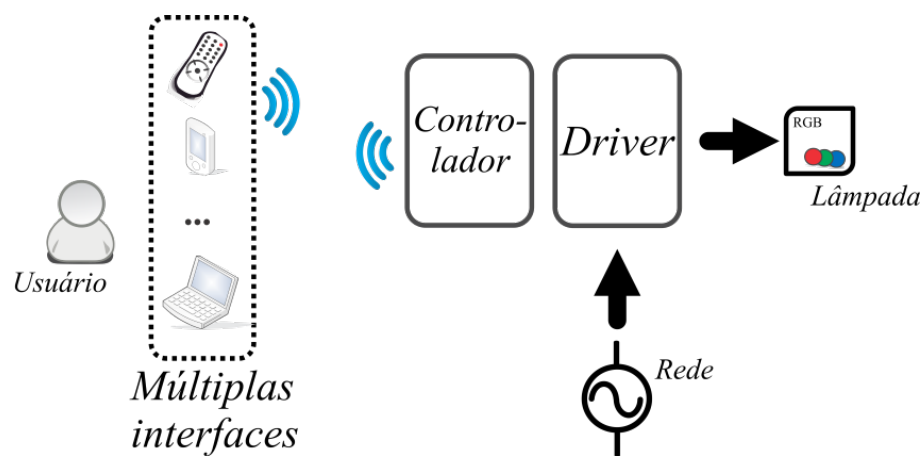


Figura 4.1: Diagrama de blocos de um sistema de gerenciamento aplicado a sistemas de iluminação com múltiplas interfaces

4.2 Circuito de acionamento dos LEDs

A fim de possibilitar a realização dos testes do sistema de gerenciamento desenvolvido, foi selecionado um circuito de acionamento e um grupo de LEDs RGB para formar a lâmpada do protótipo.

Foram utilizados três LEDs RGB de potência modelo GM-P3WRGB-30-A (GUANGMAI, 2009) ligados em série. Esse modelo de LED apresenta, em um único encapsulamento,

LEDs capazes de sintetizar as três cores primárias. As suas principais características estão apresentadas pela Tabela 4.1.

Tabela 4.1: Especificações do LED RGB de potência modelo GM-P3WRGB-30-A

	Vermelho	Verde	Azul	Unidade
Ângulo de abertura	120	120	120	°
Corrente nominal	400	350	350	mA
Tensão direta	2,2 a 2,4	3,3 a 3,6	3,3 a 3,6	V
Fluxo máximo	35	55	15	lm
Pico do espectro emitido	625	525	465	nm

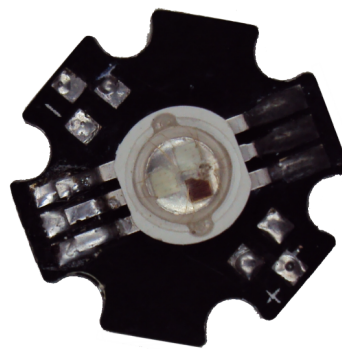


Figura 4.2: LED RGB de potência modelo GM-P3WRGB-30-A

Existem diversos circuitos que podem ser utilizados para o acionamento de LEDs, neste trabalho será utilizado um conversor linear, devido a sua fácil implementação e rápida prototipação. A maior limitação desse tipo de *driver* está no seu baixo rendimento, devido às perdas térmicas que ocorrem no regulador linear utilizado.

O circuito utilizado é formado por três reguladores lineares LM317 (ST, 2012), onde é realizada uma dimerização por modulação PWM através da comutação de um transistor ligado ao pino de regulagem de cada um dos reguladores, conforme proposto por Sá Junior (2010) e mostrado na Fig.4.3. O *driver* prototipado, bem como a lâmpada utilizada, podem ser vistos na Fig.4.4.

O valor de corrente de saída do *driver* é dado por (4.1), sendo uma relação entre a tensão no pino de regulagem (que no caso do LM317 é de 1,25V) e o valor da resistência $R1$.

$$i_{LM317} = \frac{1,25V}{R1} \quad (4.1)$$

Como o LED utilizado apresenta valores diferentes de corrente para cada uma das suas cores, é necessário projetar um valor de $R1$ para cada cor. Estes valores estão apresentados

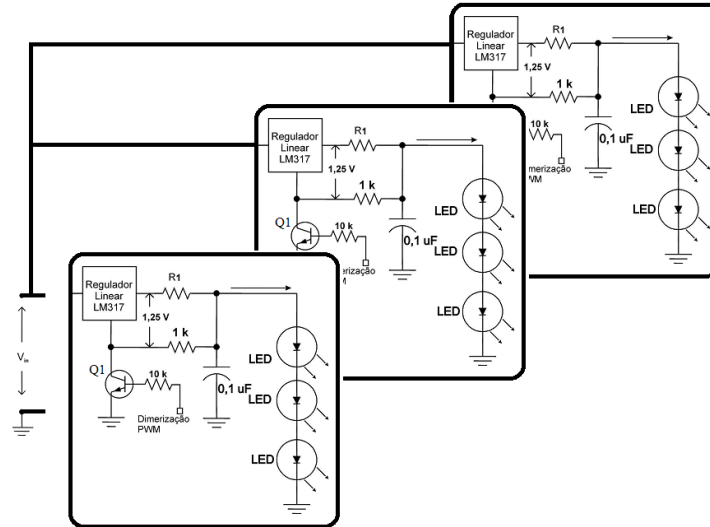


Figura 4.3: Circuito de *driver* para LEDs RGB de potência utilizado. Adaptado de (SÁ JUNIOR, 2010)

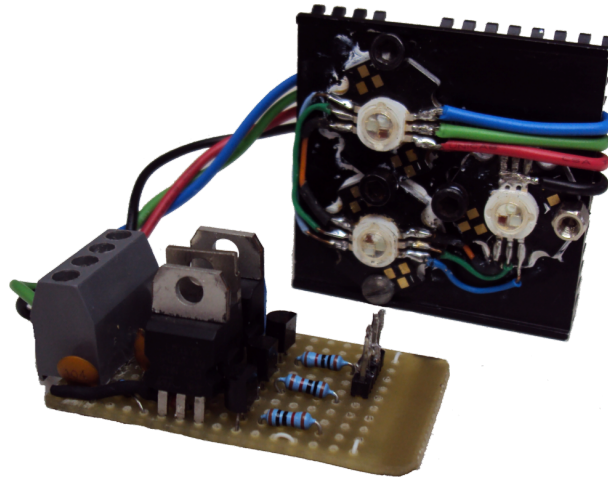


Figura 4.4: Protótipo de *driver* e lâmpada desenvolvidos

em (4.2) e (4.3).

$$R1_{\text{vermelho}} = \frac{1,25V}{0,4A} = 3,125\Omega \quad (4.2)$$

$$R1_{\text{verde}} = R_{\text{azul}} = \frac{1,25V}{0,35A} = 3,57\Omega \quad (4.3)$$

Como os valores resistências encontrados não são valores comerciais é necessário que se avalie o impacto da utilização de resistores comerciais na implementação do *driver*. Os valores de corrente obtidos são apresentados em (4.4) e (4.5).

$$i_{\text{vermelho}} = \frac{1,25V}{2\Omega + 1,2\Omega} = 390,625mA \quad (4.4)$$

$$i_{\text{verde}} = i_{\text{azul}} = \frac{1,25V}{3,6\Omega} = 347,222mA \quad (4.5)$$

A dimerização desse *driver* ocorre pela comutação do transistor $Q1$. Assim, quando o transistor não está conduzindo, a corrente projetada passa pelos LEDs. Porém com o transistor conduzindo, tem-se uma tensão de apenas $1,25V$ aplicada sobre os LEDs, como este valor é inferior ao da tensão direta dos LEDs (no caso do LED vermelho é de $6,6V$), a corrente sobre os mesmos é praticamente nula.

4.3 Controlador do sistema

Os principais pontos que precisam serem observados no desenvolvimento do controlador são:

- Identificação do protocolo mais adequado para a comunicação;
- Escolha do tipo de interface para obter uma fácil utilização;

As possíveis soluções que permitem o gerenciamento remoto, podem ser divididas em dois grupos: soluções cabeadas e soluções sem fio. A fim de se obter uma instalação fácil e de baixo custo do sistema, optou-se por utilizar uma comunicação sem fio entre os dispositivos do sistema. O uso deste tipo de comunicação permite também uma facilidade na alteração da disposição física das lâmpadas RGB.

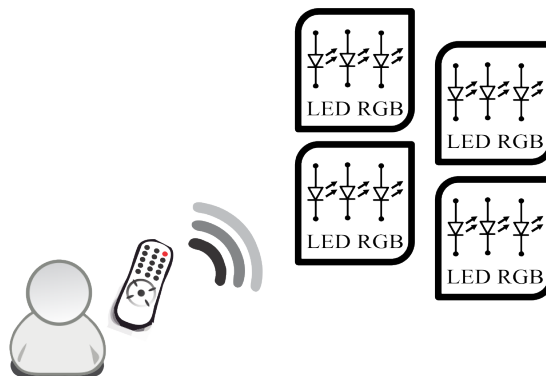


Figura 4.5: Diagrama de blocos de uma solução com controlador dedicado

Existem diversos padrões que podem ser utilizados para formar a rede sem fio, como o ZigBee[®], Bluetooth[®], Wi-Fi[™], IrDA[®] e padrões sub-GHz.

O padrão IrDA[®] tem um baixo custo, porém por ser baseado em luz infravermelha, o seu uso em ambientes externos é comprometido. O padrão Bluetooth[®] apresenta um alcance (aproximadamente 10 metros) inferior ao valor desejado, 35 metros.

Já os padrões sub-GHz, ZigBee[®] e Wi-Fi[™] são soluções que podem ser utilizadas pois atendem as especificações desse projeto. Dentro desses padrões optou-se por utilizar rádios sub-GHz, por ser uma solução de mais baixo custo.

Foram utilizados rádios AM com frequência de 433MHz, sendo que um rádio receptor Wenshing modelo RWS-375-6 (WENSHING, 2008) estará presente em cada conjunto de lâmpada (escravo) e no controlador (mestre) será utilizado um rádio transmissor Wenshing modelo TWS-DS-3 (WENSHING, 2011). Ambos os rádio utilizados podem ser vistos na Fig.4.6.

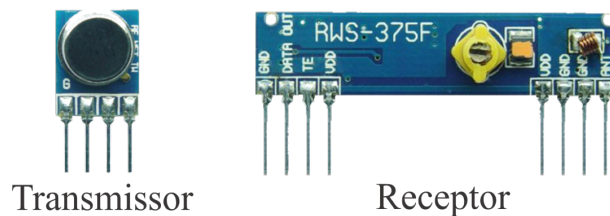


Figura 4.6: Rádios utilizados para comunicação RF433MHz, modelo TWS-DS-3 (transmissor) e RWS-375-6 (receptor)

Esse tipo de comunicação é muito utilizada quando não se necessita uma alta taxa de transferência de dados. Aplicações típicas são sistemas de alarme, de segurança e controle de portões eletrônicos. As principais especificações dos rádios utilizados são mostradas na Tabela 4.2.

Tabela 4.2: Especificações dos rádios Wenshing TWS-DS-3 e Wenshing RWS-375-6

	Transmissor (Tx)	Receptor (Rx)	Unidade
Modelo	TWS-DS-3	RWS-375-6	-
Frequência	433,92	433,92	MHz
Modulação	ASK	ASK	-
Taxa de dados	8k	4800	bps
Potência/sensibilidade	10dBm (5V)	-108	dBm
Tensão de alimentação	1,5 a 12	5	V
Alcance	80 a 100		m

Esses rádios permitem uma comunicação sem fio entre os módulos do sistema, porém esse padrão não especifica qual protocolo será utilizado nessa comunicação. Assim, neste trabalho será utilizado o protocolo desenvolvido na Seção 4.3.1.

Para permitir o teste do sistema proposto, foi desenvolvido um protótipo utilizando-se os rádios da Fig.4.6 e a plataforma de desenvolvimento Croduino (BRTOS, 2012) mostrada

na Fig.4.7. Esta plataforma foi escolhida por apresentar uma grande capacidade de personalização, facilitando assim o desenvolvimento e permitindo uma rápida prototipação. Sendo que trata-se de uma plataforma de *hardware* e *software* abertos, baseada no microcontrolador Freescale Coldfire V1 modelo MCF51JE (FREESCALE, 2010), que apresenta conectores compatíveis com *shields* Arduino. Esse microcontrolador apresenta comunicação USB, I²C, SPI e RS232 (CMOS). Entre os periféricos também estão canais PWM, DAC de 12-bit e 12 canais de ADC (12-bit).

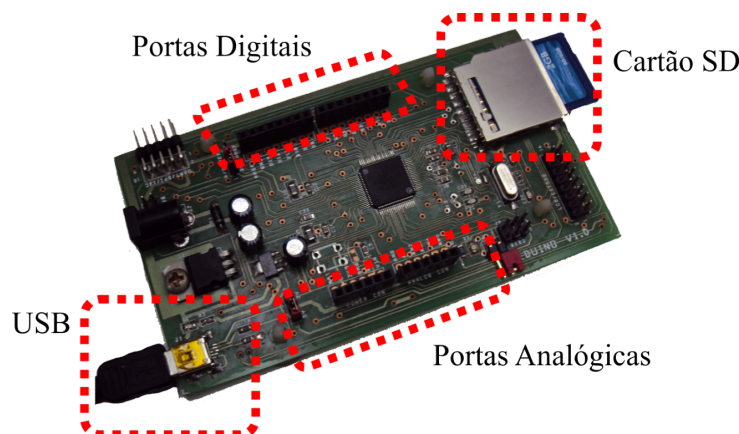


Figura 4.7: Plataforma de desenvolvimento Colduino

Tabela 4.3: Especificações da plataforma de desenvolvimento Colduino

Microcontrolador	Freescale Coldfire MCF51JE256 (50MHz)
Alimentação da plataforma	5V a 12V
Alimentação do processador	3,3V
Memória flash	256KB
Memória RAM	32KB
Tamanho	x
Conectividade	USB 2.0; RS232 (CMOS); SPI

O protótipo concebido para o transmissor pode ser visto na Fig.4.8a e para o receptor na Fig.4.8b. A interface homem máquina foi implementada através de potenciômetros, em que é possível escolher a intensidade de cada uma das três cores primárias. Outras interfaces poderiam ter sido utilizadas, como por exemplo, uma tela sensível ao toque (*touchscreen*), um teclado digital ou mesmo um botão capacitivo.

4.3.1 Protocolo Desenvolvido

Para haver a correta transmissão de dados entre os módulos da rede é necessário que esses utilizem um padrão para o envio e recebimento de pacotes, em outras palavras, é necessário utilizar um protocolo de comunicação de rede para a troca de mensagens.

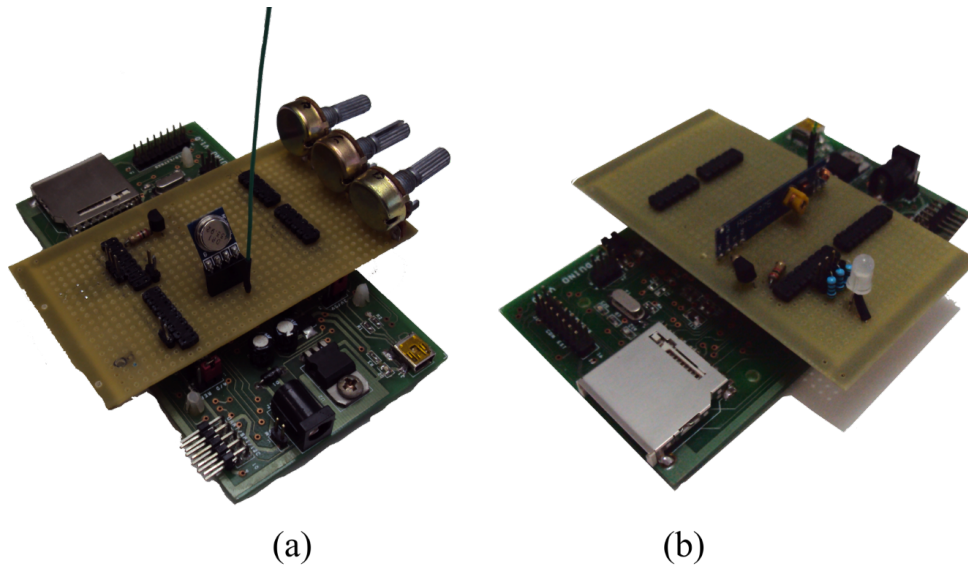


Figura 4.8: Módulos de comunicação do protótipo desenvolvido. (a) Módulo Transmissor. (b) Módulo Receptor

Os protocolos descritos anteriormente neste trabalho (Seção 3.1 e 3.2) são protocolos que apresentam muitas funcionalidades, mas também demonstraram limitações. Por exemplo, para montar um rede DALITM é necessário a utilização de rádios com capacidade de transmissão e recepção em todos os dispositivos. E redes baseadas no padrão DMX512A têm uma alta taxa de utilização do canal de transmissão.

Devido aos requisitos deste projeto, não é necessário comunicação entre todos os dispositivos da rede, sendo necessário apenas que o sistema de gerenciamento envie comandos para as lâmpadas RGB do sistema de iluminação. Assim, é possível utilizar uma topologia de rede mestre-escravo, na qual apenas o mestre tem capacidade de transmitir mensagens aos seus escravos. Porém, o mestre não recebe nenhum tipo de mensagem dos escravos.

Esta limitação na capacidade de transmissão e recepção de dados permite o uso de um *hardware* de baixo custo, uma vez que o mestre apresentará apenas um rádio transmissor e os escravos apenas um rádio receptor.

Nesta seção é exposto um novo protocolo, desenvolvido para comunicação entre mestre e escravos. Este protocolo tem a característica de ser mais simples que os outros protocolos analisados, porém isso não compromete o atendimento das necessidades deste projeto.

Como descrito na Seção 4.3, segundo o fabricante dos rádios utilizados, existe uma limitação de no máximo 4800bps . Porém, durante o desenvolvimento deste projeto observou-se que ao utilizar a taxa máxima havia um considerável número de pacotes perdidos. Portanto, foi utilizada uma taxa de transmissão de 2400bps (o dobro da taxa utilizada pelo padrão DALITM) que não apresentou problemas na transmissão e recepção

dos pacotes.

Tabela 4.4: Comparativo entre o protocolo desenvolvido e os protocolos DALITM e DMX512-A

	DALI TM	DMX512-A	Protocolo desenvolvido
Topologia de Rede	Mestre-Escravo	Mestre-Escravo	Mestre-Escravo
Comunicação	Bidirecional	Unidirecional	Unidirecional
Taxa de Dados	1200bps	250kbps	2400bps
Número de dispositivos	64	512	256
Número de comandos	256	256	256

As mensagens do protocolo desenvolvido neste trabalho iniciam com um *byte* que identifica o endereço de destino, seguido pelo comando a ser executado e eventuais dados desse comando. A estrutura padrão de mensagens enviadas pelos mestres é mostrada na Fig.4.9.

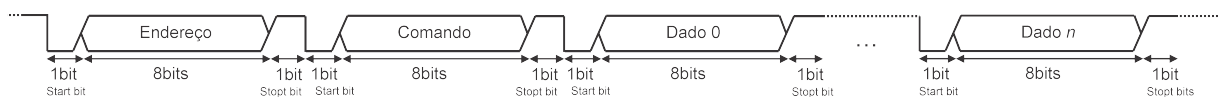


Figura 4.9: Padrão de mensagem do protocolo desenvolvido

O *byte* reservado para identificação da rede de destino do comando permite que cada mestre controle, de maneira individual, um grande número (até 256) de escravos ou grupos de escravos. Esse identificador também pode ser usado para permitir que co-existam mais de um mestre dentro do alcance dos escravos, limitado em no máximo 256 mestres diferentes.

Foi reservado um *byte* para a implementação de comandos nesse protocolo, permitindo que sejam utilizados 256 comandos diferentes. A quantidade de *bytes* de dados que vem após o comando, depende do comando enviado.

Para a aplicação desejada, gerenciamento de um sistema de iluminação decorativa, foi necessário a implementação de apenas um comando, sendo ele, o comando que permite a escolha da cor sintetizada (comando $0x24$ ou 36) pelos nós escravos. Pode parecer um número muito pequeno de comandos, mas vale ressaltar que muitas das funcionalidades presentes em protocolos mais complexos, como o DALITM, estão presentes nesse novo protocolo.

Por exemplo, o comando DALITM que permite incrementar a potência ou o que permite carregar o valor máximo de potência, podem ser realizadas no novo protocolo apenas com o uso do comando de troca de cor. Pois, como é o mestre que envia todos os comandos, este pode armazenar o último valor enviado e com isso calcular a nova cor de

maneira a se obter os mesmos efeitos dos comandos DALI™ citados. Outras funcionalidades também podem ser implementadas, a Tabela 4.5 mostra uma equivalência entre os principais comandos DALI™ e o comando do protocolo desenvolvido.

Tabela 4.5: Equivalência entre os comando do protocolo desenvolvido e comandos do padrão DALI™

Comando DALI™	Descrição	Equivalente
<i>Off</i>	Desliga instantaneamente	Troca de cor(0,0,0)
<i>Step Up</i>	Incrementa a potência	Troca de cor(Cor anterior+1)
<i>On and Step Up</i>	Liga no valor mínimo ou incrementa a potência	Troca de cor(Valor Mínimo) Troca de cor(Cor anterior+1)
<i>Step Down</i>	Decrementa potência até valor mínimo	Troca de cor(Cor anterior-1) Troca de cor(Valor Mínimo)
<i>Step Down and Off</i>	Decrementa potência até zero	Troca de cor(Cor anterior-1) Troca de cor(0,0,0)
<i>Recall MIN Level</i>	Potência no valor mínimo	Troca de cor(Valor Mínimo)
<i>Recall MAX Level</i>	Valor de potência no máximo	Troca de cor(255,255,255)

A mensagem do protocolo desenvolvido para o comando de troca de cor contém três *bytes* de dados, um para cada cor primária, conforme mostrado na estrutura de mensagem da Fig.4.10.

Cada um dos *bytes* enviados representa o valor da dimerização para cada cor RGB, ou seja, tem-se 256 pontos na dimerização. Como a relação entre o valor enviado e potência entregue à lâmpada foi implementada de maneira linear, temos uma resolução equivalente a 0,39%.

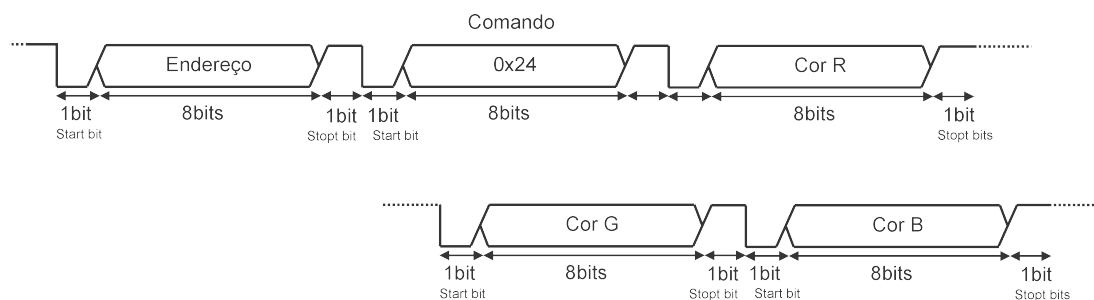


Figura 4.10: Comando para troca da cor RGB do protocolo desenvolvido

Para realizar o envio do comando foi implementada uma função no *firmware* do microcontrolador, presente no mestre, que imprime na porta serial RS232 os valores de cada pacote deste comando, conforme visto em Algoritmo 4.2.

Enquanto que, para decodificação deste protocolo no nó escravo, foi desenvolvida uma máquina de estados e a estrutura da função é mostrada no Algoritmo 4.1.

Ainda, foram desenvolvidas funções para leitura da IHM no *firmware* do nó mestre, a

Algoritmo 4.1 Função para envio da cor RGB implementada no nó mestre

Entrada: Cor Vermelha *Red*, Cor Verde *Green*, Cor Azul *Blue*

```
imprime Endereco {Envia pela serial o byte com o endereço de destino do pacote}
imprime 36 {Envia pela serial o código do comando para troca de cor}
imprime Red {Envia pela serial o byte com valor da cor vermelha}
imprime Green {Envia pela serial o byte com valor da cor verde}
imprime Blue {Envia pela serial o byte com valor da cor azul}
```

Algoritmo 4.2 Função para decodificação do protocolo implementada nos nós escravos

Entrada: Novo byte recebido pela porta serial, *NewByte***Saída:** Cor Vermelha *Red*, Cor Verde *Green*, Cor Azul *Blue**PacoteRecebido* \leftarrow *NewByte***se** *FlagEnderecoRecebido* = 0 **então** **se** *PacoteRecebido* = Endereço do escravo **então** *FlagEnderecoRecebido* \leftarrow 1 *ContadorPacotes* \leftarrow 0 **fim se****senão** **caso** (*ContadorPacotes*) **seja** 0: **se** *PacoteRecebido* = 36 **então** *ContadorPacotes* \leftarrow 1 **senão** *FlagEnderecoRecebido* \leftarrow 0 **fim se** **seja** 1: *Red* \leftarrow *PacoteRecebido* *ContadorPacotes* \leftarrow 2 **seja** 2: *Green* \leftarrow *PacoteRecebido* *ContadorPacotes* \leftarrow 3 **seja** 3: *Blue* \leftarrow *PacoteRecebido* *ContadorPacotes* \leftarrow 4 *FlagEnderecoRecebido* \leftarrow 0 *FlagPacoteRecebido* \leftarrow 1 **fim caso****fim se**

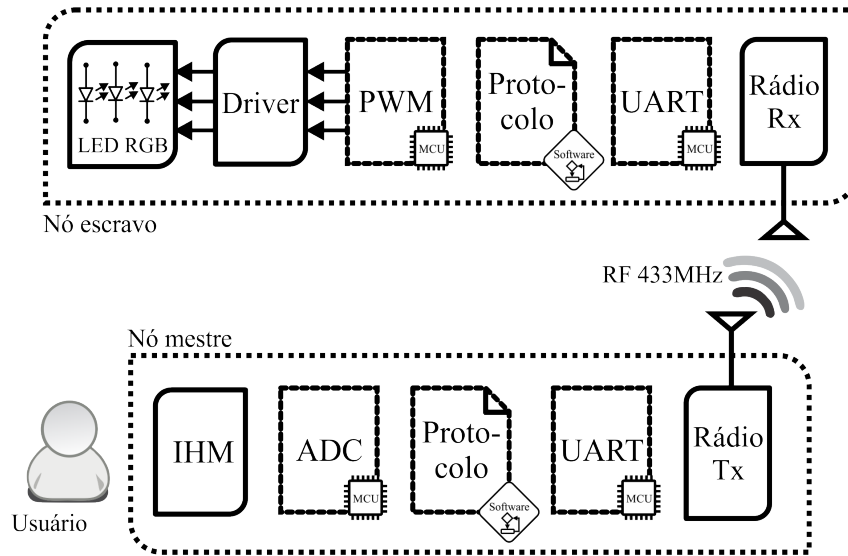


Figura 4.11: Diagrama de blocos da solução com controlador dedicado apresentada

fim de obter os parâmetros a serem transmitidos na rede. E no nó escravo foi implementada uma rotina que permite modificar o ciclo de trabalho do módulo PWM utilizado para dimerizar o *driver* dos LEDs RGB.

A Fig.4.11 mostra o diagrama de blocos do sistema de iluminação decorativa com gerenciamento remoto sem fio desenvolvido nessa seção.

4.4 Interface com dispositivos portáteis

Uma maneira de se obter uma interface de baixo custo, confiável e de fácil utilização é pela utilização de *notebooks*, *smartphones*, *tablets* e diversos outros dispositivos portáteis que já apresentam alguma comunicação sem fio, como Wi-Fi™ e Bluetooth®.

Como o alcance dos módulos Bluetooth® é de poucos metros, neste projeto será utilizado um módulo Wi-Fi™ que apresenta um alcance suficiente, taxa de dados superior à necessária e está presente na maioria dos dispositivos portáteis atuais.

Uma vez que o usuário utiliza algum dispositivo que ele já possua, os custos envolvidos nessa interface serão apenas o custo do módulo Wi-Fi™ (presente no controlador) e o custo de desenvolvimento de softwares.

A fim de reduzir o custo geral do sistema, será utilizado uma segunda rede (de mais baixo custo) para repassar os comandos para as lâmpadas RGB. Assim o usuário em seu dispositivo portátil envia um dado através da rede Wi-Fi™, que por sua vez é recebido pelo controlador (mestre) que repassa, por outra rede, os comandos aos escravos. A idéia

geral dessa topologia pode ser vista na Fig.4.12.

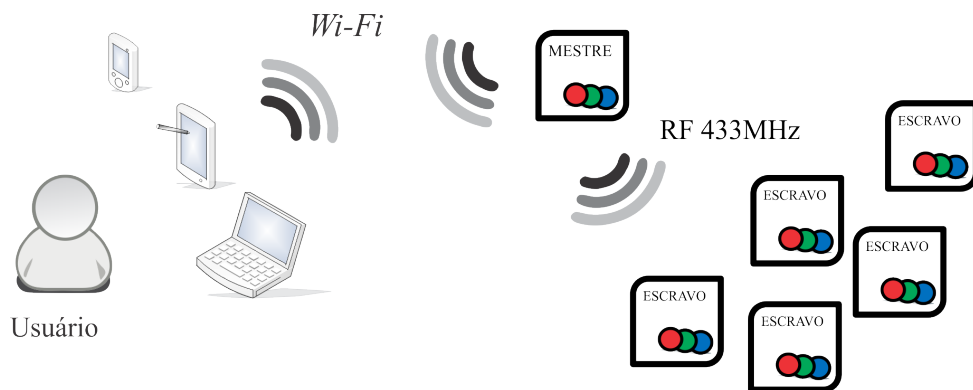


Figura 4.12: Diagrama de blocos de uma solução com interface para dispositivos móveis

4.4.1 Hardware para comunicação Wi-Fi™

Para obter a interface Wi-Fi™, o protótipo montado utilizou um roteador Wi-Fi™ TP-Link® modelo Wr703N. Este pequeno roteador ($5,7\text{cm} \times 5,7\text{cm}$) de baixo custo utiliza uma CPU Atheros AR7240 de 400MHz e apresenta uma porta USB 2.0, 4MB de memória *flash* e 32MB de memória RAM. Ainda, é possível utilizar uma porta para comunicação serial. A Fig.4.13 mostra o roteador utilizado.



Figura 4.13: Roteador Wi-Fi™ TP-Link® Modelo WR703N

Porém, para utilizar todas as funcionalidades deste *hardware*, o seu *firmware* original foi substituído por uma distribuição GNU/Linux.

Assim foi escolhida a distribuição OpenWRT (OPENWRT, 2012), desenvolvida para ser utilizada em sistemas embarcados, como roteadores Wi-Fi™.

Tabela 4.6: Especificações do roteador Wi-Fi™ TP-Link® Modelo WR703N

Processador	Atheros AR7240 (400MHz)
Alimentação do roteador	5V
Alimentação do processador	3V
Memória flash	4MB
Memória RAM	32MB
Chipset	Atheros AR9331
Potência Wi-Fi	20dBm (100mW)
Tamanho	5,7cm x 5,7cm
Conectividade	USB 2.0; RS232 (CMOS); <i>Ethernet</i>

Devido aos limitados recursos de memória destes dispositivos (que normalmente não chegam à 64MB de memória RAM e 8MB de memória *flash*), essa distribuição apresenta uma versão reduzida do Linux, ou seja, não foram implementadas todas as funcionalidades presente em versões completas, como Ubuntu e Debian.

A customização do *firmware* pode ser feita por linha de comando em um terminal serial, Telnet ou SSH. Onde é possível adicionar ou remover funcionalidades do roteador para atender as necessidades do desenvolvedor. A Fig.4.14 mostra o terminal Telnet do OpenWRT.

```

192.168.1.1 - PuTTY

=== IMPORTANT ===
Use 'passwd' to set your login password
this will disable telnet and enable SSH
-----

BusyBox v1.19.4 (2012-04-10 18:34:52 PDT) built-in shell (ash)
Enter 'help' for a list of built-in commands.

|_| W I R E L E S S   F R E E D O M
ATTITUDE ADJUSTMENT (bleeding edge, r31249) -----
* 1/4 oz Vodka      Pour all ingredients into mixing
* 1/4 oz Gin        tin with ice, strain into glass.
* 1/4 oz Amaretto
* 1/4 oz Triple sec
* 1/4 oz Peach schnapps
* 1/4 oz Sour mix
* 1 splash Cranberry juice
-----
root@(none) :/#

```

Figura 4.14: Terminal Telnet para configuração e customização do OpenWrt

As portas disponíveis para comunicação no roteador são a porta *Ethernet*, a porta USB e a porta RS232 (CMOS). Com a utilização do OpenWRT, pode-se modificar a função dos pinos da porta serial a fim de obter uma comunicação SPI. Como o microcontrola-

dor (MCF51JE) utilizado no controlador mestre não apresenta uma porta *Ethernet*, esta possibilidade foi descartada para realizar a comunicação entre o roteador e o controlador.

A principal vantagem em utilizar a porta RS232 é que este padrão de comunicação está disponível na maioria dos microcontroladores de baixo custo, o que permite o desenvolvimento futuro de uma segunda geração de controladores mestres com um custo ainda menor. A desvantagem dessa opção é que apesar de existir uma porta, não existe um conector na placa do roteador. Assim, para se obter acesso aos pinos da porta, é necessário que se abra o roteador e que seja realizada uma solda em pelo menos três pontos da placa, referência de tensão, pino de transmissão e de recepção. A Fig.4.15 mostra os pontos de transmissão (identificado como TP_OUT) e recepção (T_IN) do roteador utilizado.

Estas vantagens e desvantagens são compartilhadas com a porta SPI, uma vez que ela utiliza os mesmos pinos da porta RS232.

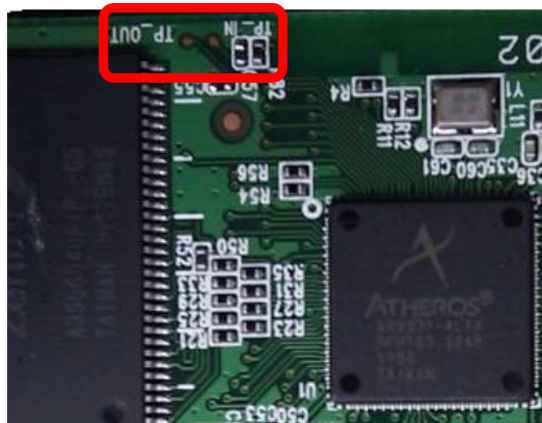


Figura 4.15: Porta RS232 do roteador Wi-Fi™ utilizado

Já a porta USB não está presente na maioria dos microcontroladores de baixo custo, mas está disponível facilmente para ser utilizada no roteador, conforme pode ser visto na Fig.4.13. Para utilizar esta comunicação, são necessários componentes adicionais na placa do microcontrolador, como um circuito de oscilador externo (cristal e capacitores), circuito para regulagem da tensão de alimentação e um conector USB.

Neste trabalho foi desenvolvida a comunicação entre o roteador e o controlador mestre utilizando a porta RS232, onde também foi utilizado o protocolo desenvolvido, porém, com uma taxa de dados de 9600bps , valor superior a apresentada anteriormente.

Para criar a interface de gerenciamento entre usuário e roteador, foi habilitado um servidor HTTP no roteador com o objetivo de servir uma página *web*. O que permite que o usuário gerencie todo o sistema pelo acesso, a partir do navegador do seu dispositivo portátil, a esta página *web*. A interface de gerenciamento desenvolvida pode ser vista na

Fig.4.16 e o código fonte da página *web* encontra-se no Anexo A.

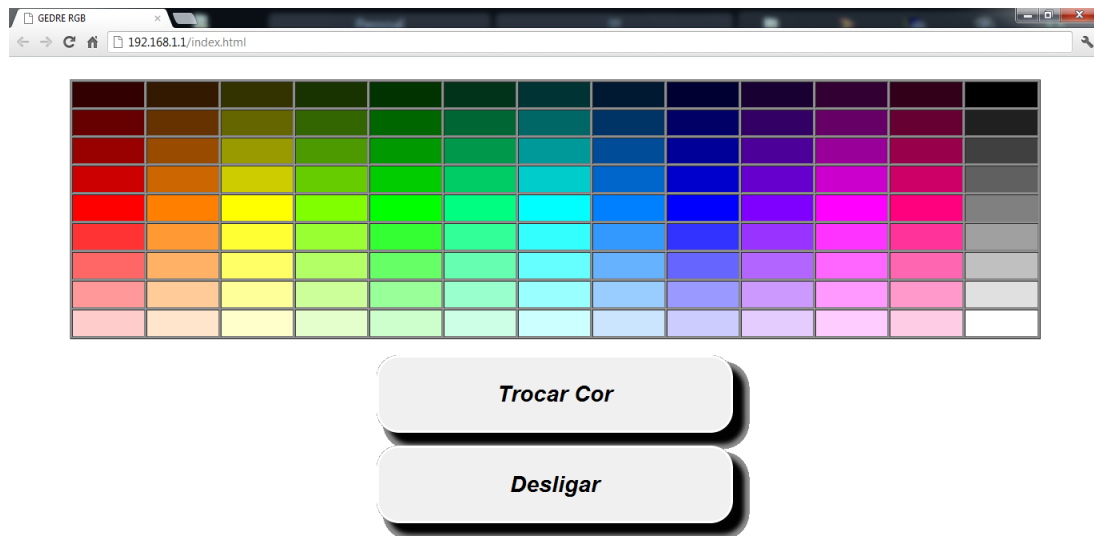


Figura 4.16: Interface de gerenciamento desenvolvida para dispositivos portáteis

Para possibilitar o envio dos dados pela porta RS232 do roteador foi escrito um *shell script* que é chamado pela página *web*, a qual passa como parâmetro uma palavra contendo as três cores primárias, e o *script* realiza a transmissão de um *byte* para cada uma das cores. O Algoritmo 4.3 representa o *script* desenvolvido para o envio de cores.

Algoritmo 4.3 *Script* utilizado para envio de cores pela porta RS232 do roteador

Entrada: *String* com cor resultante, *RGB*

Saída: Cor Vermelha *Red*, Cor Verde *Green*, Cor Azul *Blue*,

Quebra *string* com cores em três cores

imprime *Endereco*

imprime *Comando*

imprime *Red*

imprime *Green*

imprime *Blue*

Redireciona o navegador para página inicial

Assim, é possível apresentar uma solução para gerenciamento de sistemas de iluminação através da utilização de dispositivos portáteis. A Fig.4.17 mostra o diagrama de blocos desta solução.

Um protótipo de laboratório foi desenvolvido para o nó mestre, o qual apresenta uma interface Wi-Fi™ para recebimento dos comandos do usuário e uma interface subGHz para a retransmissão desses comandos para os nós escravos. Esse protótipo pode ser visto na Fig.4.18.

O protótipo desenvolvido para atuar como nó escravo (receptor) e da lâmpada (*driver*

e grupo de LEDs RGB) do sistema de iluminação decorativa é mostrado na Fig.4.19.

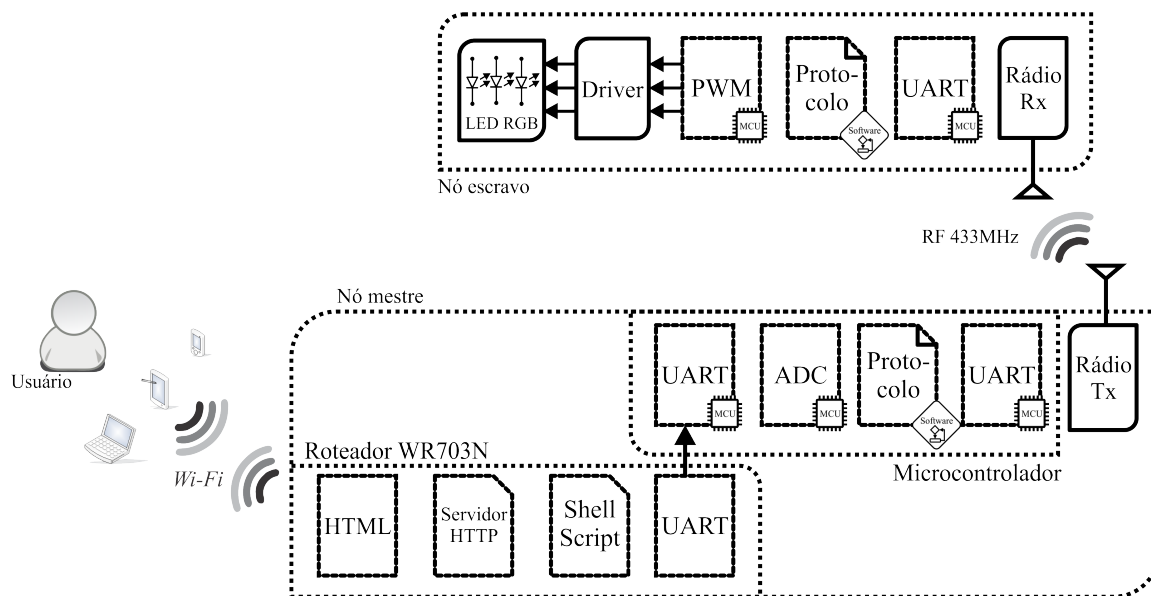


Figura 4.17: Diagrama de blocos da solução com interface para dispositivos móveis apresentada

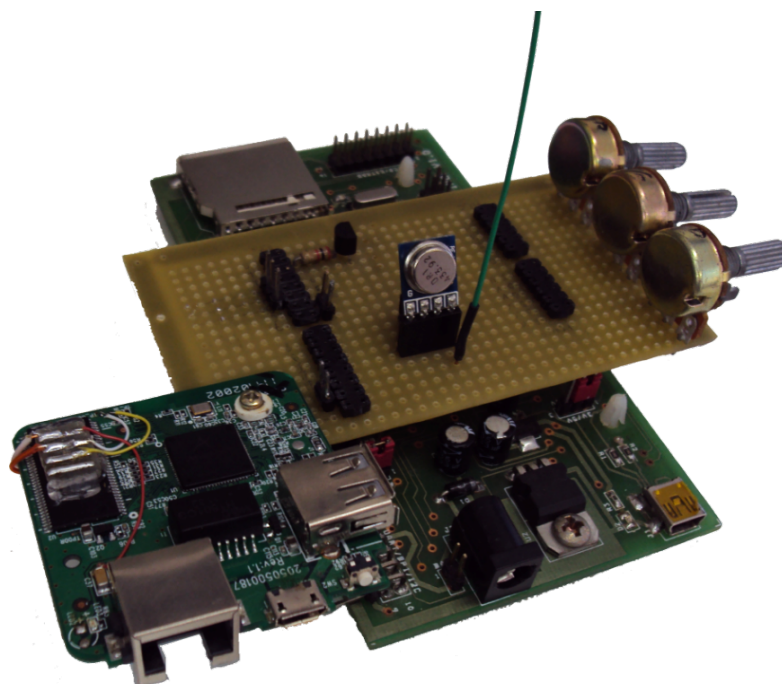


Figura 4.18: Módulo mestre com interface Wi-Fi™

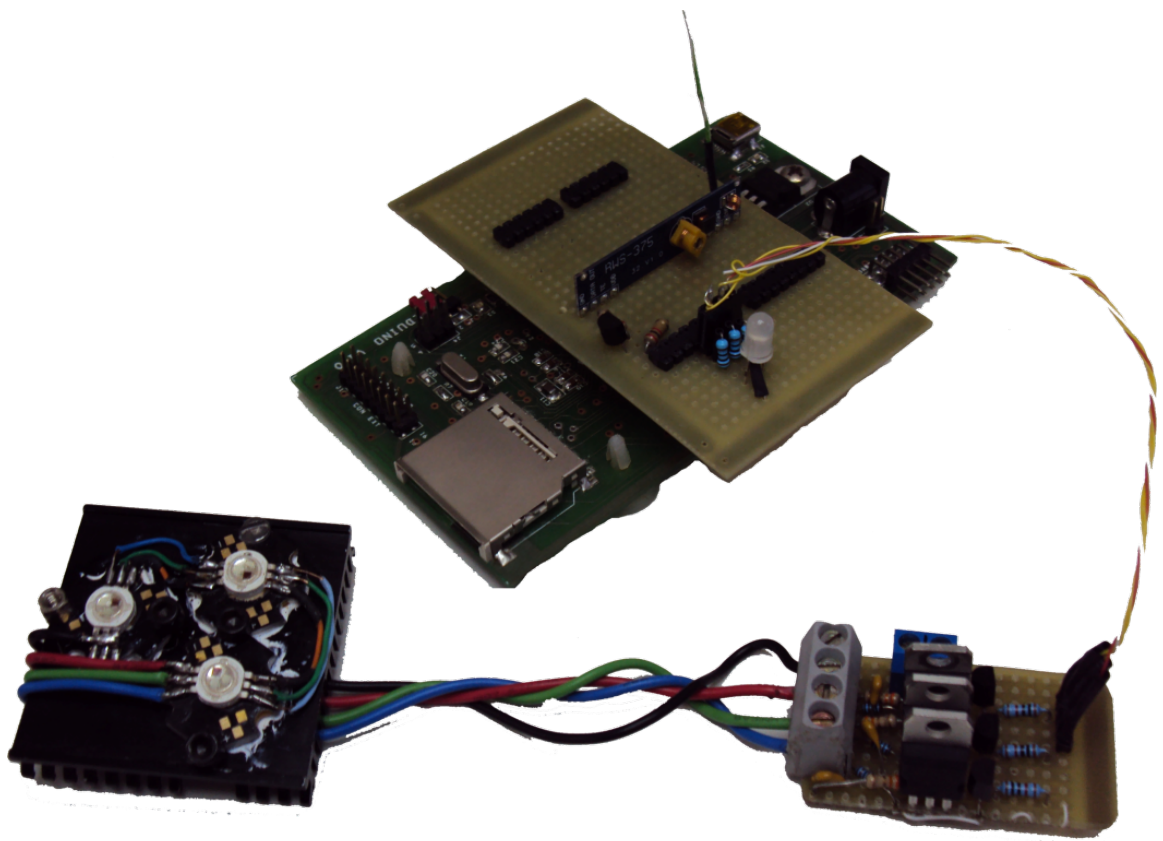


Figura 4.19: Módulo escravo - Receptor, driver e lâmpada RGB

5 *RESULTADOS EXPERIMENTAIS*

Neste capítulo são apresentados os resultados experimentais obtidos em laboratório do sistema proposto no Cap.4.

Para demonstrar o funcionamento desse sistema foram escolhidas, de maneira aleatória, intensidades distintas para cada uma das cores primárias e posteriormente foi avaliado o desempenho do sistema em sintetizar cada uma dessas três cores selecionadas no sistema de gerenciamento.

Os resultados mostrados nesta seção foram obtidos utilizando-se uma dimerização PWM de $25kHz$ e uma taxa de transmissão de dados entre mestre e escravos de $2400bps$.

5.1 Resultados para cor azul escuro

A primeira cor escolhida para ser sintetizada foi o azul escuro (*Dark Blue*), representada pelo código RGB hexadecimal `#00008B`.



Figura 5.1: Cor azul escuro

Sendo que a Fig.5.2 mostra a mensagem enviada pelo mestre (Canal 1) e a mensagem recebida pelo escravo (Canal 3) com o comando de troca de cor e dados dessa cor.

A mensagem enviada é uma sequência de cinco *bytes* com valores 35, 36, 0, 0 e 139. Onde o valor 35 representa o endereço de rede escolhido, 36 é o valor do comando de troca de cor e as cores enviadas são: 0 para cor vermelha, 0 para cor verde e 139 para a cor azul.

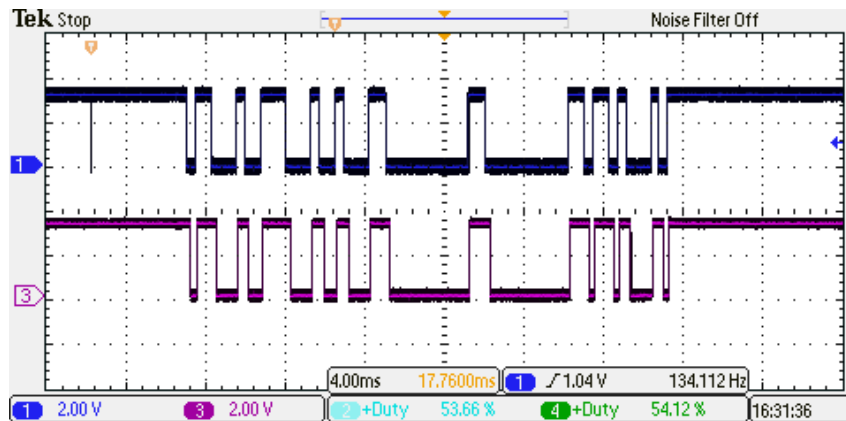


Figura 5.2: Cor azul escuro: Mensagem enviada pelo mestre (Canal 1 - $2V/div$) e recebida pelo escravo (Canal 3 - $2V/div$); Escala de tempo $17,76ms$

Cada cor é representada por um *byte* (256 possibilidades), tanto no protocolo desenvolvido, quanto no código RGB hexadecimal. A cor selecionada apresenta um valor decimal de 139 para a cor azul, ou seja, 54,3% de azul. A Fig.5.3, mostra o sinal PWM obtido no mestre (Canal 2) e no escravo (Canal 4).

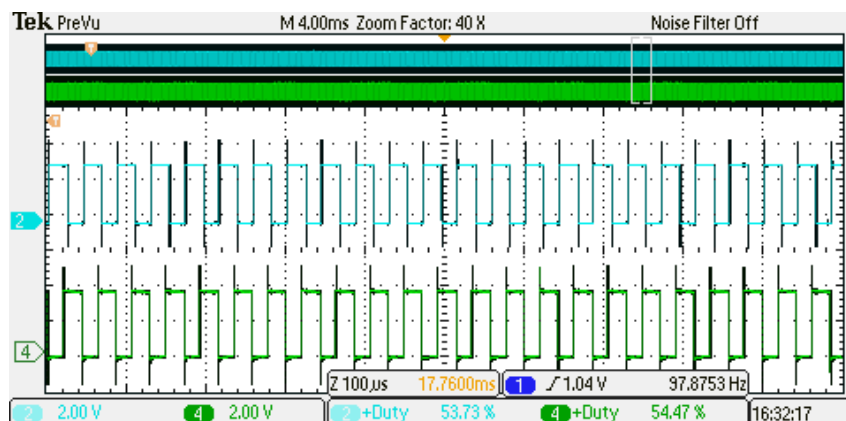


Figura 5.3: Sinal PWM para dimerização da cor azul: Nó mestre (Canal 2 - $2V/div$) e nó escravo (Canal 4 - $2V/div$); Escala de tempo $17,76ms$

Ao analisar a Fig.5.3, temos que o ciclo de trabalho do sinal PWM utilizado para dimerização da cor azul é de 53,73% para o mestre e de 54,47%, valores estes compatíveis com o desejado de 54,3%.

5.2 Resultados para cor vermelho vinho

Para testar a sintetização de cores vermelhas, escolheu-se a cor vermelho vinho (*Maroon*), esta cor é representada pelo código RGB hexadecimal #800000, ou seja, apenas 50% de vermelho.

A mensagem do sistema de gerenciamento de cores é mostrada na Fig.5.5, onde o

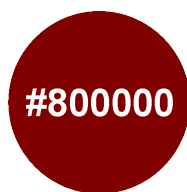


Figura 5.4: Cor vermelho vinho

Canal 1 representa a mensagem enviada pelo mestre e a mensagem recebida pelo escravo (Canal 3). Essa mensagem é composta pela sequência de *bytes*: 35, 36, 128, 0 e 0.

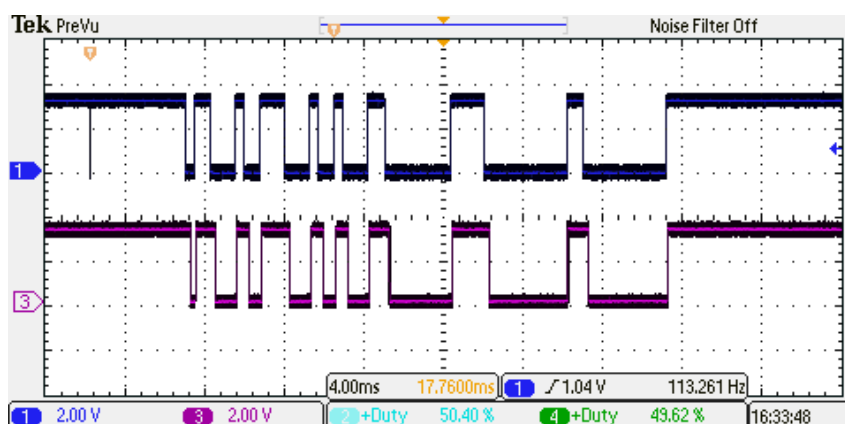


Figura 5.5: Cor vinho: Mensagem enviada pelo mestre (Canal 1 - $2V/div$) e recebida pelo escravo (Canal 3 - $2V/div$); Escala de tempo $17,76ms$

A forma de onda do canal PWM utilizado para dimerização do LED vermelho é mostrado na Fig.5.6. A figura mostra um ciclo de trabalho de aproximadamente 50% tanto para o nó mestre (Canal 2) quanto para o nó escravo (Canal 4), que é exatamente o valor selecionado pelo usuário no sistema de gerenciamento.

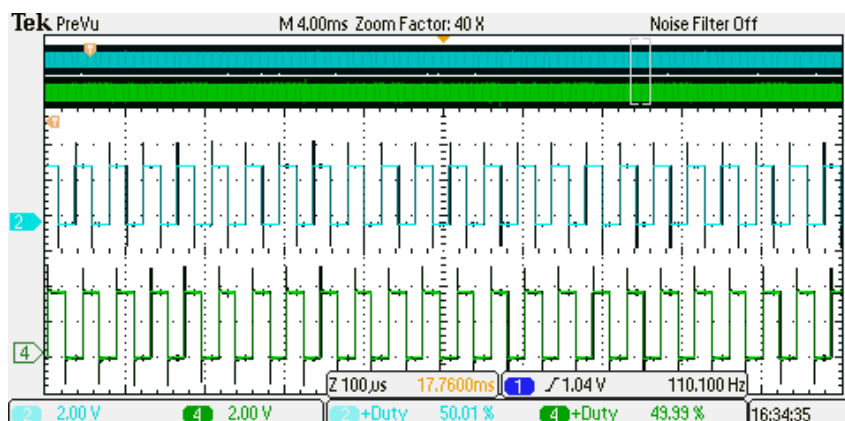


Figura 5.6: Cor vinho: Sinal PWM no mestre (Canal 2 - $2V/div$) e escravo (Canal 4 - $2V/div$); Escala de tempo $17,76ms$

5.3 Resultados para cor verde escuro

Esta seção mostra os resultados obtidos ao selecionar a cor representada pelo código RGB #006400, chamada de verde escuro (*DarkGreen*).



Figura 5.7: Cor verde escuro

A mensagem (35, 36, 0, 100 e 0) enviada pelo mestre e recebida pelo escravo é vista na Fig.5.8.

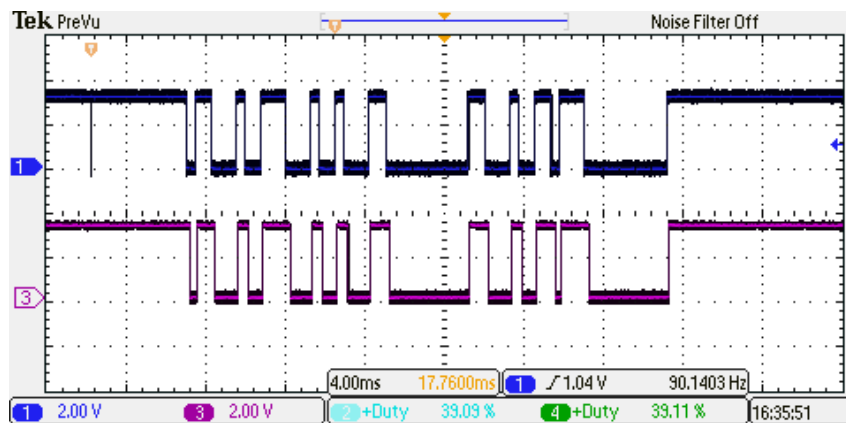


Figura 5.8: Cor verde escuro: Mensagem enviada pelo mestre (Canal 1 - 2V/div) e recebida pelo escravo (Canal 3 - 2V/div); Escala de tempo 17,76ms

Para se sintetizar esta cor, é necessário 39,1% de verde, conforme a Fig.5.9, os valores experimentais obtidos foram de 38,78% (mestre) e 39,11% no nó escravo.

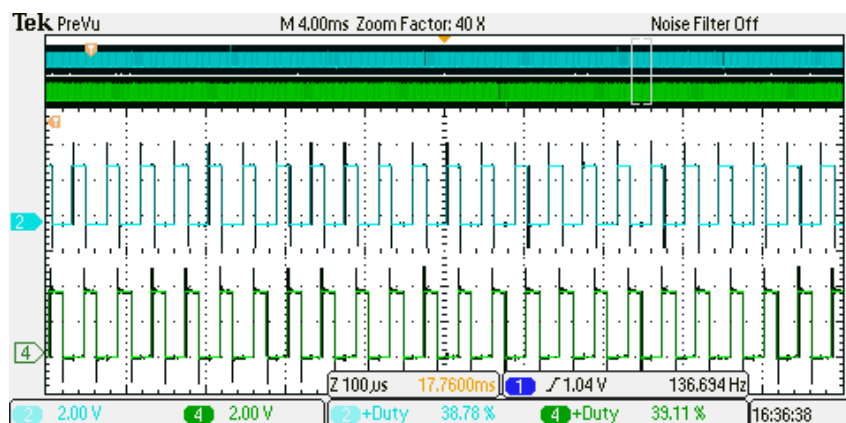


Figura 5.9: Cor verde escuro: dimerização no mestre (Canal 2 - 2V/div) e no escravo (Canal 4 - 2V/div); Escala de tempo 17,76ms

6 CONCLUSÃO

Nesse trabalho foi apresentado o estudo e desenvolvimento de um sistema de iluminação decorativa com gerenciamento remoto.

Em aplicações de iluminação decorativa é interessante que se proporcione o controle não apenas da intensidade luminosa, mas também da cor sintetizada pela lâmpada.

Assim, foi utilizado como fonte de luz um grupo de LEDs RGB de potência, pois esses apresentam capacidade de sintetização de luz em uma grande faixa do espectro visível. Além de apresentarem dimerização independente em cada uma das cores, o que os torna uma ótima alternativa para esse tipo de iluminação.

Ainda, outras características do LED, como o tamanho físico reduzido e o baixo ângulo de abertura, facilitam o desenvolvimento de luminárias para o sistema.

O sistema de gerenciamento foi desenvolvido de maneira que proporcione múltiplas interfaces entre o usuário e o sistema, sendo possível assim, controlar o sistema de iluminação através de um controlador dedicado, bem como a partir de dispositivos portáteis, como *notebooks*, *smartphones* e *tablets*.

As principais características do sistema proposto são:

- Interface intuitiva e de fácil utilização;
- Baixo custo;
- Possibilidade de uso em ambientes externos, com um alcance de até 100 metros;
- Fácil instalação do sistema.

A interface que permite o gerenciamento a partir de dispositivos portáteis foi projetada para ser simples e possibilitar sua utilização de forma intuitiva. Além disso, como o dispositivo utilizado para se obter essa interface é um roteador Wi-Fi™ que, por sua

vez, disponibiliza uma página *web*, não é necessária a instalação de nenhum aplicativo adicional no dispositivo portátil do usuário.

Nesse trabalho, também foi realizada uma análise dos protocolos de comunicação utilizados atualmente em sistemas de iluminação e posteriormente foi desenvolvido um novo protocolo, que proporciona, através de uma rede sem fio mestre-escravo de baixo custo, a comunicação entre o controlador mestre e as luminárias RGB do sistema de iluminação.

Apesar desse protocolo desenvolvido ter sido aplicado apenas para o gerenciamento de sistemas de iluminação, existe a possibilidade de ampliação do sistema, a fim de permitir outras aplicações, pois há vários comandos do protocolo que não foram implementados.

Existem várias possibilidades de aplicações, como por exemplo, a implementação de uma função que permita o controle de abertura e fechamento de cortinas permitindo assim, o gerenciamento da entrada de luz natural em ambientes internos.

Ao final deste trabalho foram apresentados os resultados obtidos a partir de um protótipo de laboratório do sistema proposto, resultados esses, que comprovam a viabilidade do sistema sendo essa uma solução de baixo custo que atende a todos os requisitos iniciais do projeto.

As principais contribuições identificadas desse trabalho são:

- A possibilidade de gerenciamento do sistema a partir de múltiplas interfaces homem máquina;
- A utilização de dispositivos portáteis como uma alternativa de interface, sem necessidade de instalação de nenhum aplicativo adicional no dispositivo do usuário;
- O desenvolvimento de um protocolo de comunicação que, apesar de simples, é capaz de atender plenamente às necessidades do sistema proposto.

6.1 Sugestões de trabalhos futuros

As principais sugestões para continuação deste trabalho encontradas pelo autor são:

1. A implementação de outros comandos no protocolo desenvolvido, a fim de ampliar as funções do sistema de gerenciamento de iluminação. Um exemplo é a implementação de um comando que permita a troca de cor com esmaecimento da luz sintetizada pelas lâmpadas;

2. A conexão entre o roteador Wi-Fi™ e o controlador mestre (dispositivo que redireciona as mensagens para as lâmpadas RGB) foi desenvolvida utilizando-se a porta RS232 do roteador, porém existem outras possibilidades, como utilizar a porta USB disponível nos dois dispositivos. Assim, um possível trabalho é a implementação dessa possibilidade, bem como uma avaliação profunda das vantagens e desvantagens de cada uma das conexões;
3. Aproveitar a disponibilidade de uma porta *Ethernet* no roteador Wi-Fi™ utilizado para ampliar ainda mais o número de interfaces possíveis, adicionando por exemplo, a possibilidade de gerenciamento remoto através da internet.

REFERÊNCIAS

- BARGHINI, A.; MEDEIRO, B. de. A iluminação artificial e o impacto sobre o meio ambiente. *Revista Brasileira de Ciências Ambientais*, n.5, p. 4–15, Dezembro 2006.
- BRTOS. *Website*. 2012. Disponível em: <<http://brtosblog.wordpress.com/>>.
- BULLOUGH, J. D. Lighting answers: Led lighting systems. *National Lighting Product Information Program, Lighting Research Center, Rensselaer Polytechnic Institute*, v. 7, 2003.
- CONTENTI, C. Digitally addressable dali dimming ballast. In: *Applied Power Electronics Conference and Exposition, 2002. APEC 2002. Seventeenth Annual IEEE*. [S.l.: s.n.], 2002. v. 2, p. 936 –942 vol.2.
- DE PAULA, J. B. *Converting RGB Images to LMS Cone Activations*. [S.l.], 2006.
- FAEBER. *Catálogo Luminária de LED Delfino*. [S.l.], 2012. Disponível em: <<http://www.faeber.com.br>>.
- FRAYTAG, J. et al. Lighting systems energy efficiency based on different human visual conditions. In: *Industry Applications (INDUSCON), 2010 9th IEEE/IAS International Conference on*. [S.l.: s.n.], 2010. p. 1 –6.
- FREESCALE. *Digitally Addressable Lighting Interface (DALI) Unit Using the MC68HC908KX8. Designer Reference Manual DRM004/D*. [S.l.], 2002.
- FREESCALE. *Folha de especificação Flexis 32-bit Microcontrollers MCF51JE256/128*. [S.l.], 2010.
- GUANGMAI, S. *Folha de especificação GM-P3WRGB-30-A*. [S.l.], 2009. Disponível em: <<http://szgmled.com/en/ProductView.asp?ID=60>>.
- HAJIABOLI, A.; POPOVIC, M. Human retinal photoreceptors: Electrodynamic model of optical microfilters. *Selected Topics in Quantum Electronics, IEEE Journal of*, v. 14, n. 1, p. 126 –130, jan.-feb. 2008. ISSN 1077-260X.
- HEIN, P. Dali-a digital addressable lighting interface for lighting electronics. In: *Industry Applications Conference, 2001. Thirty-Sixth IAS Annual Meeting. Conference Record of the 2001 IEEE*. [S.l.: s.n.], 2001. v. 2, p. 901 –905 vol.2. ISSN 0197-2618.
- IES. *Lighting Control Protocols. Technical Memorandum TM-23-11*. [S.l.], November 2011.
- LEVITON. *Folha de especificação DMX Lighting Controller 1000 Series*. [S.l.], 2006. Disponível em: <<http://www.leviton.com/>>.
- MICROCHIP. *The RS-232/DALI Bridge Interface, Application Notes AN811*. [S.l.], 2002.

- NEC. *Digital Addressable Lighting Interface(DALI). Application Notes U19884EE1V0AN00*. [S.l.], July 2009.
- NEC. *DMX512. Lighting ASSP. Application Notes U19885EE1V0AN00*. [S.l.], July 2009.
- OPENWRT. *Website*. 2012. Disponível em: <<http://openwrt.org/>>.
- OSRAM. *Catálogo Geral 2012*. [S.l.], 2012.
- OSRAM. *Monumento ao Cristo Redentor Ganha Iluminação em LED*. [S.l.], 2012. Disponível em: <<http://www.osram.com.br/appsbr/cristoredentor>>.
- PANDYA, P. *Using a PIC® Microcontroller for DMX512 Communication. Application Notes AN1076*. [S.l.], 2007.
- PHILIPS. *LUXEON Rebel White LEDs. Technical Datasheet DS64*. [S.l.], 2011. Disponível em: <<http://www.philipslumileds.com/uploads/28/DS64-pdf>>.
- PINTO, R. A. *Projeto e Implementação de Lâmpadas para Iluminação de Interiores Empregando Diodos Emissores de Luz (LEDs)*. Dissertação (Mestrado) — Universidade Federal de Santa Maria, 2009.
- RONAT, O. *The Digital Addressable Lighting Interface (DALI) : An Emerging Energy-Conserving Lighting Solution. Technical Paper TP4/9/2002*. [S.l.], 2002.
- SÁ JUNIOR, E. M. *Estudo de Estruturas de Reatores Eletrônicos para LEDs de Iluminação*. Tese (Doutorado) — Universidade Federal De Santa Catarina (UFSC), 2010.
- SCHUBERT, E. F. *Light-emitting Diodes*. [S.l.]: Cambridge University Press, 2003.
- ST. *Folha de especificação LM117, LM217 e LM317. 1.2 V to 37 V adjustable voltage regulators*. [S.l.], Fev. 2012.
- STOCKMAN, A.; SHARPE, L. The spectral sensitivities of the middle and long wavelength sensitive cones. *Vision Research*, v. 40, p. 1711 a 1737, 2000.
- STOCKMAN, A.; SHARPE, L.; FACH, C. The spectral sensitivity of the human short-wavelength sensitive cones derived from thresholds and color matches. *Vision Research*, v. 39, p. 2901 a 2927, 1999.
- TEXAS INSTRUMENTS. *Digital Addressable Lighting Interface (DALI). Application Report SLAA422*. [S.l.], November 2009.
- TRAXON. *Folha de especificação Liner Shield AC XB*. [S.l.], 2010. Disponível em: <<http://www.traxontechnologies.com>>.
- TRAXON. *Folha de especificação Wall Washer Shield AC XB-3*. [S.l.], 2010. Disponível em: <<http://www.traxontechnologies.com>>.
- TRIDONIC. *DALI Manual. Technical Information*. [S.l.], Aug 2011.
- WENSHING. *Folha de especificação RWS-375 RF Module Series*. [S.l.], Abril 2008.
- WENSHING. *Folha de especificação TWS-DS RF Module Series*. [S.l.], Maio 2011.

ZHANG, Y.; ZHOU, P.; WU, M. Research on dali and development of master-slave module. In: *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*. [S.l.: s.n.], 2006. p. 1106 –1110.

APÊNDICE A - CÓDIGO HTML IMPLEMENTADO

Listing A.1: index.html

```
1 <html lang="pt-br" dir="ltr">
  <head>
    <meta http-equiv="Content-Type" content="text/html;_charset=utf-8"
      />
    <meta http-equiv="Content-Language" content="pt-br,_pt">
    <meta http-equiv="pragma" content="No-cache">
6    <meta name="author" content="Vargas" />
    <title>GEDRE RGB</title>
  </head>

11 <body bgcolor="#909090">
  <br>
  <center>
    <!-- Tabela de cores -->
    <table border="1" width="90%" height="50%">
16      <tr>
        <td bgcolor="#330000" onClick="SetColor('#330000')">&nbsp;</td>
        <td bgcolor="#331900" onClick="SetColor('#331900')">&nbsp;</td>
        <td bgcolor="#333300" onClick="SetColor('#333300')">&nbsp;</td>
        <td bgcolor="#193300" onClick="SetColor('#193300')">&nbsp;</td>
21      <td bgcolor="#003300" onClick="SetColor('#003300')">&nbsp;</td>
        <td bgcolor="#003319" onClick="SetColor('#003319')">&nbsp;</td>
        <td bgcolor="#003333" onClick="SetColor('#003333')">&nbsp;</td>
        <td bgcolor="#001933" onClick="SetColor('#001933')">&nbsp;</td>
        <td bgcolor="#000033" onClick="SetColor('#000033')">&nbsp;</td>
26      <td bgcolor="#190033" onClick="SetColor('#190033')">&nbsp;</td>
        <td bgcolor="#330033" onClick="SetColor('#330033')">&nbsp;</td>
        <td bgcolor="#330019" onClick="SetColor('#330019')">&nbsp;</td>
        <td bgcolor="#000000" onClick="SetColor('#000000')">&nbsp;</td>
```

```
31     </tr>
32     <tr>
33         <td bgcolor="#660000" onClick="SetColor ('#660000') " >&nbsp;</td>
34         <td bgcolor="#663300" onClick="SetColor ('#663300') " >&nbsp;</td>
35         <td bgcolor="#666600" onClick="SetColor ('#666600') " >&nbsp;</td>
36         <td bgcolor="#336600" onClick="SetColor ('#336600') " >&nbsp;</td>
37         <td bgcolor="#006600" onClick="SetColor ('#006600') " >&nbsp;</td>
38         <td bgcolor="#006633" onClick="SetColor ('#006633') " >&nbsp;</td>
39         <td bgcolor="#006666" onClick="SetColor ('#006666') " >&nbsp;</td>
40         <td bgcolor="#003366" onClick="SetColor ('#003366') " >&nbsp;</td>
41         <td bgcolor="#000066" onClick="SetColor ('#000066') " >&nbsp;</td>
42         <td bgcolor="#330066" onClick="SetColor ('#330066') " >&nbsp;</td>
43         <td bgcolor="#660066" onClick="SetColor ('#660066') " >&nbsp;</td>
44         <td bgcolor="#660033" onClick="SetColor ('#660033') " >&nbsp;</td>
45         <td bgcolor="#202020" onClick="SetColor ('#202020') " >&nbsp;</td>
46     </tr>
47     <tr>
48         <td bgcolor="#990000" onClick="SetColor ('#990000') " >&nbsp;</td>
49         <td bgcolor="#994C00" onClick="SetColor ('#994C00') " >&nbsp;</td>
50         <td bgcolor="#999900" onClick="SetColor ('#999900') " >&nbsp;</td>
51         <td bgcolor="#4C9900" onClick="SetColor ('#4C9900') " >&nbsp;</td>
52         <td bgcolor="#009900" onClick="SetColor ('#009900') " >&nbsp;</td>
53         <td bgcolor="#00994C" onClick="SetColor ('#00994C') " >&nbsp;</td>
54         <td bgcolor="#009999" onClick="SetColor ('#009999') " >&nbsp;</td>
55         <td bgcolor="#004C99" onClick="SetColor ('#004C99') " >&nbsp;</td>
56         <td bgcolor="#000099" onClick="SetColor ('#000099') " >&nbsp;</td>
57         <td bgcolor="#4C0099" onClick="SetColor ('#4C0099') " >&nbsp;</td>
58         <td bgcolor="#990099" onClick="SetColor ('#990099') " >&nbsp;</td>
59         <td bgcolor="#99004C" onClick="SetColor ('#99004C') " >&nbsp;</td>
60         <td bgcolor="#404040" onClick="SetColor ('#404040') " >&nbsp;</td>
61     </tr>
62     <tr>
63         <td bgcolor="#CC0000" onClick="SetColor ('#CC0000') " >&nbsp;</td>
64         <td bgcolor="#CC6600" onClick="SetColor ('#CC6600') " >&nbsp;</td>
65         <td bgcolor="#CCCC00" onClick="SetColor ('#CCCC00') " >&nbsp;</td>
66         <td bgcolor="#66CC00" onClick="SetColor ('#66CC00') " >&nbsp;</td>
67         <td bgcolor="#00CC00" onClick="SetColor ('#00CC00') " >&nbsp;</td>
68         <td bgcolor="#00CC66" onClick="SetColor ('#00CC66') " >&nbsp;</td>
69         <td bgcolor="#00CCCC" onClick="SetColor ('#00CCCC') " >&nbsp;</td>
70         <td bgcolor="#0066CC" onClick="SetColor ('#0066CC') " >&nbsp;</td>
71         <td bgcolor="#0000CC" onClick="SetColor ('#0000CC') " >&nbsp;</td>
72         <td bgcolor="#6600CC" onClick="SetColor ('#6600CC') " >&nbsp;</td>
73         <td bgcolor="#CC00CC" onClick="SetColor ('#CC00CC') " >&nbsp;</td>
```



```

    <td bgcolor="#CC0066" onClick="SetColor('#CC0066')">&nbsp;</td>
    <td bgcolor="#606060" onClick="SetColor('#606060')">&nbsp;</td>
</tr>
76 <tr>
    <td bgcolor="#FF0000" onClick="SetColor('#FF0000')">&nbsp;</td>
    <td bgcolor="#FF8000" onClick="SetColor('#FF8000')">&nbsp;</td>
    <td bgcolor="#FFFF00" onClick="SetColor('#FFFF00')">&nbsp;</td>
    <td bgcolor="#80FF00" onClick="SetColor('#80FF00')">&nbsp;</td>
81 <td bgcolor="#00FF00" onClick="SetColor('#00FF00')">&nbsp;</td>
    <td bgcolor="#00FF80" onClick="SetColor('#00FF80')">&nbsp;</td>
    <td bgcolor="#00FFFF" onClick="SetColor('#00FFFF')">&nbsp;</td>
    <td bgcolor="#0080FF" onClick="SetColor('#0080FF')">&nbsp;</td>
    <td bgcolor="#0000FF" onClick="SetColor('#0000FF')">&nbsp;</td>
86 <td bgcolor="#7F00FF" onClick="SetColor('#7F00FF')">&nbsp;</td>
    <td bgcolor="#FF00FF" onClick="SetColor('#FF00FF')">&nbsp;</td>
    <td bgcolor="#FF007F" onClick="SetColor('#FF007F')">&nbsp;</td>
    <td bgcolor="#808080" onClick="SetColor('#808080')">&nbsp;</td>
</tr>
91 <tr>
    <td bgcolor="#FF3333" onClick="SetColor('#FF3333')">&nbsp;</td>
    <td bgcolor="#FF9933" onClick="SetColor('#FF9933')">&nbsp;</td>
    <td bgcolor="#FFFF33" onClick="SetColor('#FFFF33')">&nbsp;</td>
    <td bgcolor="#99FF33" onClick="SetColor('#99FF33')">&nbsp;</td>
96 <td bgcolor="#33FF33" onClick="SetColor('#33FF33')">&nbsp;</td>
    <td bgcolor="#33FF99" onClick="SetColor('#33FF99')">&nbsp;</td>
    <td bgcolor="#33FFFF" onClick="SetColor('#33FFFF')">&nbsp;</td>
    <td bgcolor="#3399FF" onClick="SetColor('#3399FF')">&nbsp;</td>
    <td bgcolor="#3333FF" onClick="SetColor('#3333FF')">&nbsp;</td>
101 <td bgcolor="#9933FF" onClick="SetColor('#9933FF')">&nbsp;</td>
    <td bgcolor="#FF33FF" onClick="SetColor('#FF33FF')">&nbsp;</td>
    <td bgcolor="#FF3399" onClick="SetColor('#FF3399')">&nbsp;</td>
    <td bgcolor="#A0A0A0" onClick="SetColor('#A0A0A0')">&nbsp;</td>
</tr>
106 <tr>
    <td bgcolor="#FF6666" onClick="SetColor('#FF6666')">&nbsp;</td>
    <td bgcolor="#FFB266" onClick="SetColor('#FFB266')">&nbsp;</td>
    <td bgcolor="#FFFF66" onClick="SetColor('#FFFF66')">&nbsp;</td>
    <td bgcolor="#B2FF66" onClick="SetColor('#B2FF66')">&nbsp;</td>
111 <td bgcolor="#66FF66" onClick="SetColor('#66FF66')">&nbsp;</td>
    <td bgcolor="#66FFB2" onClick="SetColor('#66FFB2')">&nbsp;</td>
    <td bgcolor="#66FFFF" onClick="SetColor('#66FFFF')">&nbsp;</td>
    <td bgcolor="#66B2FF" onClick="SetColor('#66B2FF')">&nbsp;</td>
    <td bgcolor="#6666FF" onClick="SetColor('#6666FF')">&nbsp;</td>

```

```

116     <td bgcolor="#B266FF" onClick="SetColor ('#B266FF') " >&nbsp;</td>
        <td bgcolor="#FF66FF" onClick="SetColor ('#FF66FF') " >&nbsp;</td>
        <td bgcolor="#FF66B2" onClick="SetColor ('#FF66B2') " >&nbsp;</td>
        <td bgcolor="#C0C0C0" onClick="SetColor ('#C0C0C0') " >&nbsp;</td>
</tr>
121 <tr>
        <td bgcolor="#FF9999" onClick="SetColor ('#FF9999') " >&nbsp;</td>
        <td bgcolor="#FFCC99" onClick="SetColor ('#FFCC99') " >&nbsp;</td>
        <td bgcolor="#FFFF99" onClick="SetColor ('#FFFF99') " >&nbsp;</td>
        <td bgcolor="#CCFF99" onClick="SetColor ('#CCFF99') " >&nbsp;</td>
126 <td bgcolor="#99FF99" onClick="SetColor ('#99FF99') " >&nbsp;</td>
        <td bgcolor="#99FFCC" onClick="SetColor ('#99FFCC') " >&nbsp;</td>
        <td bgcolor="#99FFFF" onClick="SetColor ('#99FFFF') " >&nbsp;</td>
        <td bgcolor="#99CCFF" onClick="SetColor ('#99CCFF') " >&nbsp;</td>
        <td bgcolor="#9999FF" onClick="SetColor ('#9999FF') " >&nbsp;</td>
131 <td bgcolor="#CC99FF" onClick="SetColor ('#CC99FF') " >&nbsp;</td>
        <td bgcolor="#FF99FF" onClick="SetColor ('#FF99FF') " >&nbsp;</td>
        <td bgcolor="#FF99CC" onClick="SetColor ('#FF99CC') " >&nbsp;</td>
        <td bgcolor="#E0E0E0" onClick="SetColor ('#E0E0E0') " >&nbsp;</td>
</tr>
136 <tr>
        <td bgcolor="#FFCCCC" onClick="SetColor ('#FFCCCC') " >&nbsp;</td>
        <td bgcolor="#FFE5CC" onClick="SetColor ('#FFE5CC') " >&nbsp;</td>
        <td bgcolor="#FFFFCC" onClick="SetColor ('#FFFFCC') " >&nbsp;</td>
        <td bgcolor="#E5FFCC" onClick="SetColor ('#E5FFCC') " >&nbsp;</td>
141 <td bgcolor="#CCFFCC" onClick="SetColor ('#CCFFCC') " >&nbsp;</td>
        <td bgcolor="#CCFFE5" onClick="SetColor ('#CCFFE5') " >&nbsp;</td>
        <td bgcolor="#CCFFFF" onClick="SetColor ('#CCFFFF') " >&nbsp;</td>
        <td bgcolor="#CCE5FF" onClick="SetColor ('#CCE5FF') " >&nbsp;</td>
        <td bgcolor="#CCCCFF" onClick="SetColor ('#CCCCFF') " >&nbsp;</td>
146 <td bgcolor="#E5CCFF" onClick="SetColor ('#E5CCFF') " >&nbsp;</td>
        <td bgcolor="#FFCCFF" onClick="SetColor ('#FFCCFF') " >&nbsp;</td>
        <td bgcolor="#FFCCE5" onClick="SetColor ('#FFCCE5') " >&nbsp;</td>
        <td bgcolor="#FFFFFF" onClick="SetColor ('#FFFFFF') " >&nbsp;</td>
</tr>
151 </table>

<!-- Envio da cor selecionada -->
156 <br>
<form name="EnviaCorHardForm" action="http://192.168.1.1:502/cgi-bin/
        setcolor.cgi" onsubmit="return_validateForm ()">

```

```

161 <input type="hidden" id="RCV" name="RCV" size="2" maxlength=2>
    <input type="hidden" id="GCV" name="GCV" size="2" maxlength=2>
    <input type="hidden" id="BCV" name="BCV" size="2" maxlength=2>
    <input type="submit" id="BotaoEnviaHard" value="Envia_Cor" style="
        height:15%;width:33%;font-size:4ex;font-style:italic;_font-weight:
        bold;border:_solid_ white;border-radius:_1em;_box-shadow:black_0.5
        em_0.5em_0.3em">
    </form>

166 <!-- Comando para desligar o sistema -->
    <form name="EnviaCorSoftForm" action="http://192.168.1.1:502/cgi-bin/
        turnoff.cgi" onsubmit="return_validateForm()">
        <input type="submit" id="BotaoTurnOff" value="Desligar" style="height
            :15%;width:33%;font-size:4ex;font-style:italic;_font-weight:bold;
            border:_solid_ white;border-radius:_1em;_box-shadow:black_0.5em_
            0.5em_0.3em">
        </form>

171 </center>
    </body>
</html>

```

```

//Script Java para troca da cor dos botões e quebra da string
<script type="text/javascript">
3  function GetHex(texto)
    {
        numero = parseInt(texto,16);
        return numero;
    }
8
function SetColor(NewColor)
{
    this.className='hov';
    var iLen = String(NewColor).length;
13  NewColor = String(NewColor).substring(iLen, iLen - 6);

    //Inverte cor selecionada:
    CorInvertida = (GetHex(NewColor)^16777215).toString(16);

18  //Quebra R,G,B e coloca no form:
    Red = (GetHex(String(NewColor).substring(0,2))).toString(16);
    Green =(GetHex(String(NewColor).substring(2,4))).toString(16);

```

```
Blue = (GetHex(String(NewColor).substring(4,6))).toString(16);  
  
23 //Coloca valor no form:  
document.getElementById("RCV").value=Red;  
document.getElementById("GCV").value=Green;  
document.getElementById("BCV").value=Blue;  
  
28 //Troca cor de fundo e da fonte dos botões do form:  
document.getElementById("BotaoEnviaHard").style.backgroundColor=  
    NewColor;  
document.getElementById("BotaoEnviaHard").style.color= CorInvertida;  
  
    }  
33 </script >
```