



**CENTRO UNIVERSITÁRIO DE BRASÍLIA
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS
CURSO DE ENGENHARIA DA COMPUTAÇÃO**

LUCAS REHEM DE AZEVEDO

AUTOMAÇÃO DE ILUMINAÇÃO RESIDÊNCIAL VIA IPAD/IPHONE

Orientadora: Prof^ª. M.C. Maria Marony Sousa Farias

BRASÍLIA

2º SEMESTRE DE 2012

LUCAS REHEM DE AZEVEDO

AUTOMAÇÃO DE ILUMINAÇÃO RESIDENCIAL VIA IPAD/IPHONE

**Trabalho apresentado ao UniCEUB – Centro
Universitário de Brasília como pré-requisito pa-
ra obtenção de Certificação de Conclusão do
Curso de Engenharia de Computação.
Orientadora: Prof^a. M.C. Maria Marony Sousa
Farias Nascimento.**

BRASÍLIA

2º SEMESTRE DE 2012

LUCAS REHEM DE AZEVEDO

AUTOMAÇÃO DE ILUMINAÇÃO RESIDENCIAL VIA IPAD/IPHONE

Trabalho apresentado ao UniCEUB – Centro Universitário de Brasília como pré-requisito para obtenção de Certificação de Conclusão do Curso de Engenharia de Computação. Orientadora: Prof^a. M.C. Maria Marony Sousa Farias Nascimento.

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas - FATECS.

Prof. Abiezer Amarília Fernandez
Coordenador do Curso

Banca Examinadora:

Prof^a. Maria Marony Sousa Farias, mestre em Engenharia Elétrica – UFPB – PB.
Orientadora

Prof^o José Julimá Bezerra Junior, mestre em Eng^a Elétrica – Instituto Militar de Engenharia Rio de Janeiro - RJ

Prof. Antônio Barbosa Júnior, especialista em engenharia de software

Prof. Luís Cláudio Lopes de Araujo, Uni-CEUB

Dedico a todos meus amigos e familiares que participaram dessa caminhada.

AGRADECIMENTOS

Primeiro a Deus. A toda minha família e amigos por todo apoio e paciência. A minha orientadora Maria Marony Sousa Farias pelas críticas e sugestões, aos meus amigos e companheiros de curso Ana Gabriela Ungierowicz, Caio de Bem, Flavio Lenzi, Bruno Queiroz, Jefferson Santos, Diogo Dantas, Lucas Mesquita, Emerson da Hora, Fillipe Galiza, Paulo Henrique, Rapahel Palmer e José Carlos. Gostaria de agradecer também o apoio dos professores Francisco Javier e Luciano Duque.

Não quero ser o melhor, mas ter a certeza de sempre dar o meu melhor

Autor

RESUMO

Este trabalho apresenta um dispositivo de controle e automação de iluminação residencial via Ipad/Iphone. Esse projeto tem por objetivo a automatização da iluminação de uma residência, provendo segurança, comodidade, acessibilidade, economia e facilidade para seus moradores. Com o intuito de facilitar o controle da iluminação da residência foi optado por utilizar um *smartphone/tablet* (Iphone/Ipad) para que a pessoa possa controlar a sua casa de qualquer lugar onde tenha conexão com a rede. Para isso, são necessários, além do *tablet* iPad, com sua interface gráfica personalizada, um servidor web, localizado em um computador, que receberá e interpretará as decisões recebidas via rede Wi-Fi e, via roteador as enviará para um *hardware* microcontrolador *Arduino* que, por sua vez, realizará a função desejada de acender ou apagar as luzes de um determinado cômodo da maquete residencial.

Palavras-chave: Automação, Iluminação, Arduino Uno, Iphone, Ipad.

ABSTRACT

This paper presents a device automation and control home lighting via Ipad / Iphone. This project aims at automating the lighting of a residence, providing safety, comfort, accessibility, ease and economy for its residents. In order to facilitate the control of lighting the residence was chosen to use a smartphone / tablet (Iphone / Ipad) so that the person can control your home from anywhere you have network connection. For this, we need, besides the iPad tablet, with its custom graphical interface, a web server, located on a computer that receives and interprets decisions received via Wi-Fi, and via the router sends to an Arduino microcontroller hardware that in turn, will perform the desired function to turn on or off lights in a particular room of the model home.

Keywords: Automation, Lighting, Arduino Uno, Iphone, Ipad.

SUMÁRIO

| | |
|---|-------------|
| AGRADECIMENTOS..... | IV |
| RESUMO | VI |
| ABSTRACT | VII |
| SUMÁRIO..... | VIII |
| LISTA DE FIGURAS..... | XI |
| LISTA DE TABELAS..... | XII |
| LISTA DE ABREVIATURAS E SIGLAS | XIII |
| CAPÍTULO 1 - INTRODUÇÃO..... | 14 |
| 1.1 Introdução ao Tema Proposto..... | 14 |
| 1.2 Motivação | 14 |
| 1.3 Objetivos..... | 15 |
| 1.4 Metodologias | 16 |
| 1.5 Estrutura da Monografia..... | 16 |
| CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA..... | 17 |
| 2.1 Segurança | 17 |
| 2.2 Conforto | 17 |
| 2.3 Acessibilidade | 18 |
| 2.4 Economia de Energia | 18 |

| | |
|---|-----------|
| CAPÍTULO 3 – REFERENCIAL TEÓRICO | 19 |
| 3.1 Automação Residencial | 19 |
| 3.2 Iluminação Residencial | 20 |
| 3.3 Microcontrolador | 21 |
| 3.3.1 Microcontrolador Arduino Uno | 21 |
| 3.3.2 Shield Ethernet | 22 |
| 3.4 Comunicação..... | 23 |
| 3.4.1 Wi-Fi..... | 23 |
| 3.4.2 UDP | 23 |
| 3.5 Wamp | 23 |
| 3.5.1 PHP..... | 24 |
| 3.6 Smartphone | 24 |
| 3.6.1 Iphone/Ipad | 24 |
| 3.7 Visão Geral do Projeto..... | 25 |
| | |
| CAPÍTULO 4 – DESCRIÇÃO DO HARDWARE E SOFTWARE..... | 26 |
| 4.1 Microcontrolador Arduino Uno..... | 26 |
| 4.1.1 Especificações..... | 26 |
| 4.1.2 Pinagem do Arduino Uno..... | 27 |
| 4.2 Relé..... | 28 |
| 4.3 Roteador Linksys WRT54G..... | 28 |
| 4.4 Arduino Software | 29 |
| 4.5 Programa PHP | 29 |
| | |
| CAPÍTULO 5 – IMPLEMENTAÇÃO..... | 30 |
| 5.1 – Fluxograma Geral do Sistema..... | 31 |

| | | |
|---|--|-----------|
| 5.2 | Elaboração do código fonte | 31 |
| 5.2.1 | Para o microcontrolador Arduino Uno | 31 |
| 5.2.2 | Para o servidor web | 34 |
| 5.3 | Elaboração do circuito | 36 |
| 5.3.1 | Placa Relé..... | 36 |
| 5.3.2 | Montagem do Circuito..... | 37 |
| CAPÍTULO 6 – RESULTADOS OBTIDOS | | 39 |
| 6.1 | Simulações | 39 |
| 6.2 | Problemas Encontrados | 39 |
| 6.3 | Orçamento do Projeto | 39 |
| CAPÍTULO 7 – CONSIDERAÇÕES FINAIS | | 41 |
| 7.1 | Conclusões | 41 |
| 7.2 | Propostas para Trabalhos Futuros..... | 41 |
| REFERÊNCIAS BIBLIOGRÁFICAS..... | | 42 |
| APÊNDICE A - CÓDIGO FONTE DO DISPOSITIVO | | 43 |
| APÊNDICE B - CÓDIGO FONTE WEB..... | | 46 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 3.1 – Micro controlador Arduino Uno..... | 22 |
| Figura 3.2 – Shield Ethernet Wiz5100..... | 22 |
| Figura 3.4 – Diagrama Esquemático do projeto proposto | 25 |
| Figura 4.1 – Micro Controlador Arduino Uno | 27 |
| Figura 4.2 – Micro Controlador Arduino..... | 27 |
| Figura 4.2 – Placa de interface de relés..... | 28 |
| Figura 4.3 – Roteador Wireless | 29 |
| Figura 5.1 – Protótipo Concluído..... | 30 |
| Figura 5.2– Fluxograma Geral do Sistema | 31 |
| Figura 5.3– Código Fonte Parte I | 32 |
| Figura 5.4– Código Fonte Parte II | 33 |
| Figura 5.5– Código Fonte Parte III | 33 |
| Figura 5.6– Código Fonte Parte IV | 34 |
| Figura 5.7– Código Fonte Parte V | 35 |
| Figura 5.8– Código Fonte Parte VI | 35 |
| Figura 5.8– Código Fonte Parte VI | 36 |
| Figura 5.9– Desenvolvimento Placa Relé..... | 37 |
| Figura 5.10– Montagem Circuito..... | 38 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 3.1– Crescimento Iphone/Ipad no mercado..... | 25 |
| Tabela 4.1– Especificação do Microcontrolador Arduino Uno..... | 26 |
| Tabela 6.1– Orçamento Projeto | 40 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-----------------|---|
| Aureside | Associação Brasileira de Automação Residencial |
| GND | <i>Graduated Neutral Density Filter</i> |
| ICSP | In Circuit Serial Programming |
| PHP | PHP Hypertext Preprocessor |
| TCP/IP | Transmission Control Protocol / Internet Protocol |
| UDP | User Datagram Protocol |
| V | Volts |
| WAMP | Windows, Apache, MySQL e PHP |
| WLAN | Wireless Local Area Network |

CAPÍTULO 1 - INTRODUÇÃO

1.1 Introdução ao Tema Proposto

Com o desenvolvimento e a facilidade de acesso às novas tecnologias, adicionado ao crescimento econômico do país, a população vê como necessidade aumentar o seu conforto e segurança. O controle e automação residencial têm aparecido como uma tendência no cenário mundial atual. Por apresentar um custo relativamente alto para a realidade da maioria da população o controle e automação residencial ainda não estão difundidos.

Muito se fala sobre as "novidades" tecnológicas que terão os lares no futuro. Como qualquer novidade, a automação residencial inicialmente é percebida pelo cliente como um símbolo de *status* e modernidade. No momento seguinte, o conforto e a conveniência por ela proporcionados passam a ser decisivos. Por fim, ela se tornará uma necessidade vital e um fator de economia. Um ótimo exemplo foi à evolução da utilização dos celulares.

Esse sentido o desenvolvimento destas ideias e sua propagação têm aumentado entre os profissionais. Assim estando todos mais preparados para absorver a demanda desse emergente mercado e participando ativamente do seu crescimento.

1.2 Motivação

Hoje conseguimos ter um melhor controle do nosso dia-a-dia com uma simples interação com nossos celulares ou computadores. Sendo assim por que não aproveitar e conseguir controlar a nossa casa pelos mesmos meios. E além da parte de comodidade, podem contar também o aumento de segurança. Seja desligando e ligando o alarme, abrindo e fechando portas ou mesmo controlando a iluminação da sua casa, simulando que pessoas estão por lá enquanto viajam.

Uma das aplicações relacionadas a controle e automação de residências, diz respeito à iluminação residencial. Muitos dispositivos atualmente têm a função de coibir indivíduos mal intencionados que, com a aproximação, fazem com que as luzes do local se acendam.

Outra potencial aplicação é a acessibilidade que pessoas deficientes têm para se movimentar ou caso queiram ligar ou desligar uma luz. Para esse segmento da população, este sistema não é apenas uma questão de conforto, e sim algo que leva a sua independência assim sem depender de ajuda para ter esse tipo de controle.

A abordagem a ser utilizada seria a instalação de uma central de controle e automação de iluminação residencial, podendo automatizar a iluminação da casa, tais como: luz da garagem, sala, quartos e jardim.

1.3 Objetivos

O projeto tem como objetivo geral a elaboração de um dispositivo de controle e automação de iluminação residencial, que tem como a principal funcionalidade, através da interação com uma página *web* e dispositivo móvel, que o proprietário possa ligar ou desligar uma luz da sua casa, e saber o estado da iluminação dos cômodos controlada por ele.

O objetivo específico deste trabalho é apresentar um protótipo de uma casa com iluminação de alguns cômodos controlada e automatizada pelo dispositivo objeto de estudo. Com o uso do micro controlador Arduino Uno em conjunto com o com um servidor apache e uma página *web* recebendo comandos de um dispositivo *mobile*. O servidor *web* se comunica com o micro controlador via *web* e dessa forma um sinal é tratado pelo dispositivo, que liga ou desliga a iluminação do cômodo. Logo que o sinal é enviado à página *web* informa que tal cômodo da residência está com a luz acesa ao proprietário.

1.4 Metodologias

Para a elaboração deste projeto, foram realizadas pesquisas bibliográficas em livros e *sites* conceituados da internet, bem como a realização de diversos testes para a análise dos componentes eletrônicos a serem utilizados no dispositivo.

1.5 Estrutura da Monografia

Esta monografia está dividida em sete capítulos, incluindo a INTRODUÇÃO, que trata da introdução ao tema proposto, à motivação do projeto, os principais objetivos, metodologias de elaboração e pesquisa.

No segundo capítulo, APRESENTAÇÃO DO PROBLEMA, é apresentada uma descrição aprofundada sobre as questões motivacionais do trabalho, as soluções existem atualmente sobre o tema proposto, e aborda de forma resumida, os benefícios do dispositivo proposto e suas restrições.

No terceiro capítulo, REFERENCIAL TEÓRICO, trata de assuntos como automação residencial, microcontroladores, e conceitos utilizados. Nesse capítulo é apresentada também uma visão geral do projeto.

No quarto capítulo, DESCRIÇÃO DO HARDWARE E SOFTWARE, aborda as especificações dos dispositivos utilizados, é detalhada a especificação dos componentes e *softwares*.

No quinto capítulo, IMPLEMENTAÇÃO, são apresentadas etapas necessárias para compreensão geral da implementação do projeto.

No sexto capítulo, RESULTADOS OBTIDOS, aborda as simulações que tiveram como objetivo testar todas as funcionalidades propostas do dispositivo, de forma a simular um ambiente real, bem como as dificuldades encontradas.

No sétimo capítulo, CONSIDERAÇÕES FINAIS, são apresentadas a conclusão e as sugestões para trabalhos futuros.

CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA

Este capítulo tem como finalidade detalhar questões motivacionais do trabalho, como segurança e comodidade residencial, que foram apresentadas resumidamente na seção — Motivação — do capítulo anterior.

São apresentadas também algumas soluções existentes para essas questões, no que diz respeito à iluminação residencial. E por fim, são demonstrados os benefícios do dispositivo proposto pelo trabalho e suas restrições.

2.1 Segurança

No ambiente de automação residencial os equipamentos de vigilância e monitoramento eletrônico podem ser considerados como sistema de segurança tendo como foco o aumento da segurança e poder ser monitorado remotamente por seus moradores.

Por meio de uma central de controle integrada ao sistema de segurança, é possível monitorar movimentos, seja com sensores de presença ou seja com alarmes monitorando a distância.

2.2 Comodidade

Além da segurança, as pessoas estão em busca de melhorar a sua qualidade de vida. Ligar a luz da garagem, para verificar se está tudo nos conformes, do jardim ou mesmo do quarto.

Em busca desta comodidade e outras, e adicionada à evolução tecnológica surgiram os dispositivos que permitem o controle e a automação de rotinas e tarefas de uma casa, não só o controle da iluminação, como de muitos outros dispositivos, que serão apresentados na seção — Automação Residencial — do próximo capítulo.

2.3 Acessibilidade

A criação de sistemas que interagem com voz, ou mesmo *touch screen* facilitam a vida de pessoas com necessidades especiais. E com o crescimento da demanda, mesmo que pequeno, vem crescendo com o tempo.

“Nos Estados Unidos, são aproximadamente 5 milhões de residências automatizadas e um mercado de US\$ 1,6 bilhão de dólares em 1998 à US\$ 3,2 bilhões para o ano de 2002 e previsão de 10,5 bilhões em 2008. No Brasil, estima-se um potencial de 2 milhões de residências apenas para o estado de São Paulo e faturamento de US\$ 100 milhões em 2004.”(Alguns aspectos sobre a Automação Residencial - Doméstica. Dissertação submetida à Universidade Federal de Santa Catarina, 2002, TEZA)

Para a população que possui algum tipo de deficiência de locomoção, esses sistemas são ferramentas necessárias para que o indivíduo consiga a sua independência, dando a possibilidade de personalizar esses sistemas como: som, televisores, iluminação, portões, internet entre vários outros.

2.4 Economia de Energia

A energia deve ser utilizada apenas quando necessária.

“A ecologia (responsabilidade ambiental) está em alta e todos estão buscando soluções viáveis para economizar energia e ao mesmo tempo ajudar o meio ambiente. Como a iluminação pode representar até 40% de uma conta de fornecimento de energia elétrica, trabalhar bem com ela pode ser um ótimo meio de começar a economizar. Dentre as mais eficientes e agradáveis opções estão os sistemas de controle de iluminação. Os produtos para controle de iluminação trazem retorno ao investimento feito além de propiciar benefícios adicionais como conforto, conveniência e segurança” (<http://www.aureside.org.br/ilum.pdf>, 2007, AURESIDE)

CAPÍTULO 3 – REFERENCIAL TEÓRICO

Para o melhor entendimento e desenvolvimento do projeto, é necessário o estudo de conceitos teóricos pertinentes ao projeto desenvolvido. Embora alguns dos conceitos citados demandem um detalhamento mais extenso do tema para sua total compreensão, visando manter o foco principal, apenas as características mais relacionadas ao projeto são apresentadas.

Nesse capítulo é apresentada também uma visão de forma geral do projeto proposto. Por isso é fundamental para o entendimento do projeto a compreensão dos tópicos abordados neste capítulo.

3.1 Automação Residencial

Conforme a definição do dicionário, Automação é o “funcionamento de uma máquina ou grupo de máquinas que, sob o controle de um programa único, permite efetuar uma série de operações.” (<http://www.dicio.com.br/automacao/>, 2012, acesso em 05/10/2012)

A automação residencial visa modificar a visão que existe hoje a respeito do controle que uma pessoa tem sob sua residência. Com uma abordagem diferenciada de como, por exemplo, acender uma luz, ligar a irrigação do jardim ou mesmo controlar outros diversos tipos de objetos como:

- a) Iluminação;
- b) Eletrodomésticos;
- c) Vigilância e alarmes;
- d) *Home theater*;
- e) Ar condicionado;
- f) Cortinas/portas automáticas;
- g) Telefonia.

No passado, o alto custo de instalação de equipamentos para automação residencial afastava mesmo quem tinha dinheiro para investir, tornando o uso de tecnologia dentro de casa uma excentricidade. Segundo estimativa da Associação Brasileira de Automação Residencial¹ (Aureside), os preços de equipamentos e a instalação caíram pela metade nos últimos quatro anos. (LEAL, 2011).

“O objetivo da automação residencial é integrar iluminação, entretenimento, segurança, telecomunicações, aquecimento, ar condicionado e muito mais através de um sistema inteligente programável e centralizado. Como consequência fornece praticidade, segurança, conforto e economia para o dia a dia dos usuários.” (ABREU, 2003).

3.2 Iluminação Residencial

Para a realização desse projeto foi tido como foco à parte de automação residencial abordando iluminação.

Thomas Edson jamais imaginaria que sua criação se tornaria tão versátil. Ele apenas queria algo que nos ajudasse a enxergar no escuro. Enquanto isso, nossas lâmpadas, podem quando ligadas a sistemas especiais de controle de iluminação, prover muitas outras utilidades. Sistemas inteligentes de iluminação podem aguçar os detalhes arquitetônicos de uma sala ou criar um clima romântico ou festivo. Ligando e desligando automaticamente, podem proteger uma casa de intrusos, fazendo-a parecer ocupada na ausência de seus proprietários. Economia de eletricidade é outra vantagem, pois a intensidade de luz é regulada conforme a necessidade e as lâmpadas não precisam ficar totalmente acesas como acontece normalmente.

¹ A Associação Brasileira de Automação Residencial (Aureside) tem como missão fomentar a adoção de tecnologias de automação residencial no país, homologar produtos e serviços na área, manter cursos de capacitação, formação e certificação profissional em automação residencial, divulgar artigos técnicos, realizar cursos e palestras na área, dentre outras missões.

3.3 Microcontrolador

“O microcontrolador é um dispositivo semicondutor em forma de circuito integrado, que integra as partes básicas de um microcomputador - microprocessador, memórias não-voláteis e voláteis e portas de entrada e saída. Geralmente, é limitado em termos de quantidade de memória, principalmente no que diz respeito à memória de dados, é utilizada em aplicações específicas, ou seja, naquelas que não necessitam armazenar grandes quantidades de dados, como automação residencial, automação predial, automação industrial e automação embarcada.” (GIMENEZ, 2005, p. 4).

São vários os fornecedores de microcontroladores. Os principais, em termos de volume de vendas no Brasil, são: (PAIOTTI, 2009).

- Microchip Technology Inc. (<http://www.microchip.com>);
- Intel Corporation (<http://www.intel.com>);
- Atmel Corporation (<http://www.atmel.com>); e
- Texas Instruments (<http://www.ti.com>);

3.3.1 Microcontrolador Arduino Uno

Para controlar a iluminação da residência que está sendo abordada na monografia foi utilizado o microcontrolador Arduino Uno. Foi o escolhido para essa implementação por possuir uma interface amigável e muitas documentações.

O Uno difere de todas as placas antecessoras no sentido de não utilizar o chip FTDI para conversão do sinal serial. Utiliza no seu lugar um Atmega8U2 programado como conversor de USB para serial.

"Uno" quer dizer um em italiano e é utilizado para marcar o lançamento do Arduino 1.0. O Uno e a versão 1.0 serão as versões de referência do Arduino, daqui para diante. O UNO é o mais recente de uma série de placas Arduino, e o modelo de referência para a plataforma Arduino.”(<http://arduino.cc/en/Main/ArduinoBoardUno>, 2012, acessado em 04/10/2012)

Sua linguagem de programação é a Wiring, linguagem com a mesma sintaxe C/C++.

A figura 3.1 ilustra uma imagem do microcontrolador Arduino Uno.

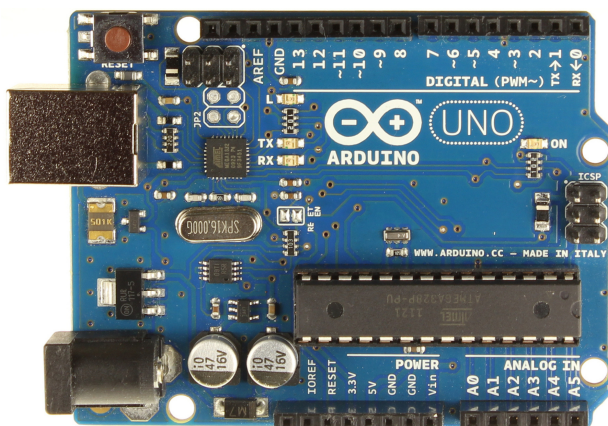


Figura 3.1 – Micro controlador Arduino Uno
Fonte: Arduino CC, disponível em http://arduino.cc/en/uploads/Main/ArduinoUno_R3_Front.jpg acesso em 04/10/2012

3.3.2 Shield Ethernet

Como a solução proposta não aborda a utilização de cabos ou fios para o manuseio do usuário final, foi necessário acoplar ao microcontrolador Arduino Uno, conforme descrição no item 3.3.1, um dispositivo que tinha como objetivo realizar a comunicação entre servidor web e microcontrolador.

O Shield Ethernet permite conectividade via Internet para projetos com Arduino. O controlador utilizado foi o Wiz5100, pois possuía uma alta recomendação dos desenvolvedores, e pode suportar até quatro conexões TCP e UDP.

A figura 3.2 ilustra uma imagem do Shield Ethernet.



Figura 3.2 – Shield Ethernet Wiz5100
Fonte: Hu infinito, disponível em <http://huinfinito.com.br/628-925-thickbox/shield-ethernet-rede-wiznet-w5100.jpg> acesso em 04/10/2012

3.4 Comunicação

Para a comunicação ser realizada foi necessário o apoio de um protocolo e uma tecnologia.

3.4.1 Wi-Fi

“Wi-Fi é um conjunto de especificações para redes locais sem fio (WLAN - Wireless Local Area Network) baseada no padrão IEEE 802.11. O nome Wi-Fi é tido como uma abreviatura do termo inglês "Wireless Fidelity", embora a Wi-Fi Alliance, entidade responsável principalmente pelo licenciamento de produtos baseados na tecnologia, nunca tenha afirmado tal conclusão.” (<http://www.infowester.com/wifi.php>, acesso em 05/10/2012)

3.4.2 UDP

O UDP (*user data protocol*) foi o protocolo utilizado para a transmissão dos pacotes entre o iPad e o computador, isso se dá pelo fato do aplicativo usado no iPad enviar os pacotes desta maneira.

O conjunto de protocolos da Internet também abrange um protocolo de transporte sem conexão, o UDP. Este oferece uma forma de as aplicações enviarem datagramas IP brutos encapsulados sem que seja necessário estabelecer uma conexão. (TANENBAUM, A. S, 1997)

3.5 Wamp

“O Servidor Wamp é um ambiente de desenvolvimento web. Ele permite que sejam criadas aplicações web com Apache2, PHP e uma base de dados MySQL.”(<http://www.wampserver.com/en/>, 2012, acessado em 05/10/2012). A partir desse software livre foram utilizados neste projeto os conceitos de servidor web e linguagem PHP.

3.5.1 PHP

É uma linguagem de programação, que sua sigla significa "*PHP: Hypertext Preprocessor*", que é utilizada para desenvolvimento Web podendo ser mesclada com código HTML. Parecida com Perl, Java e C é uma linguagem que possui uma curva de aprendizado pequena. Seu objetivo é permitir que páginas sejam geradas dinamicamente e rapidamente por desenvolvedores. http://www.php.net/manual/pt_BR/preface.php, acessado em 01/11/2012)

3.6 Smartphone

Para controlar todo o circuito que foi montado o objeto escolhido foi o Smartphone. Muito mais do que aparelhos usados para comunicação, os celulares possuem funcionalidades que o ser humano já não consegue ignorar. MP3, fotos, vídeos, internet, GPS e agenda são algumas das utilidades mais utilizadas e procuradas em um celular pelos consumidores.

Com a chegada da terceira geração (3G) os celulares ganharam novas tecnologias e novos conceitos. Agora indispensáveis para a vida das pessoas, os produtos contam com aplicativos e funcionalidades inseridos pelas empresas, que visam unir o conforto e a praticidade para satisfazer o usuário.

O Smartphone – telefone inteligente, em tradução literal – surgiu de uma combinação entre celulares e agendas eletrônicas (palmtops e handhelds). Eles possuem uma tecnologia mais avançada e seu próprio sistema operacional é escrito em código aberto, o que significa que qualquer pessoa pode simplesmente desenvolver programas que funcionam neste sistema operacional. Os principais sistemas operacionais usados são: Symbian, Blackberry OS, Windows Mobile, iOS, Sansumg Bada, Palm WebOS, Android e Maemo.

3.6.1 Iphone/Ipad

Estudos feitos pela empresa Morgan Stanley mostram que tanto o iPad quanto o Iphone vem se tornando os dispositivos móveis mais populares da história. Uma

recente pesquisa feita pela companhia “Morgan Stanley” comprova que mesmo com a disputa do mercado com outros dispositivos móveis, o crescimento continua maior do que o esperado.

A tabela 3.1 ilustra esse crescimento:

Tabela 3.1– Crescimento Iphone/Ipad no mercado

| Pesquisa iPhone e iPad | | | | |
|------------------------|-------------------|------------|------------|---------------------|
| | Pesquisa Estimada | Estimativa | Variação % | Estimativa nas ruas |
| 4Q iPhone | 31-36M | 30M | 3-20% | 28M |
| 1Q iPhone | 41M | 28M | 46% | 26M |
| 2012 iPhone | 190M | 134M | 42% | 127M |
| 2012 iPad | 81M | 52M | 56% | 54M |

FONTE: Morgan Stanley

3.7 Visão Geral do Projeto

Após leitura e compreensão dos tópicos abordados neste capítulo, já é possível o entendimento teórico do projeto proposto. O projeto basicamente é constituído microcontrolador (Arduino do modelo Uno), Shield Ethernet (compatível com arduino uno e chip 5100), relés de duas posições, servidor apache controlando uma página web em php, lâmpadas comuns de 220 V e demais componentes eletrônicos, como resistores, capacitores, oscilador de cristal, dentre outros.

O diagrama esquemático do projeto proposto pode ser observado na Figura 3.4, na qual representa de forma objetiva a composição geral do projeto.

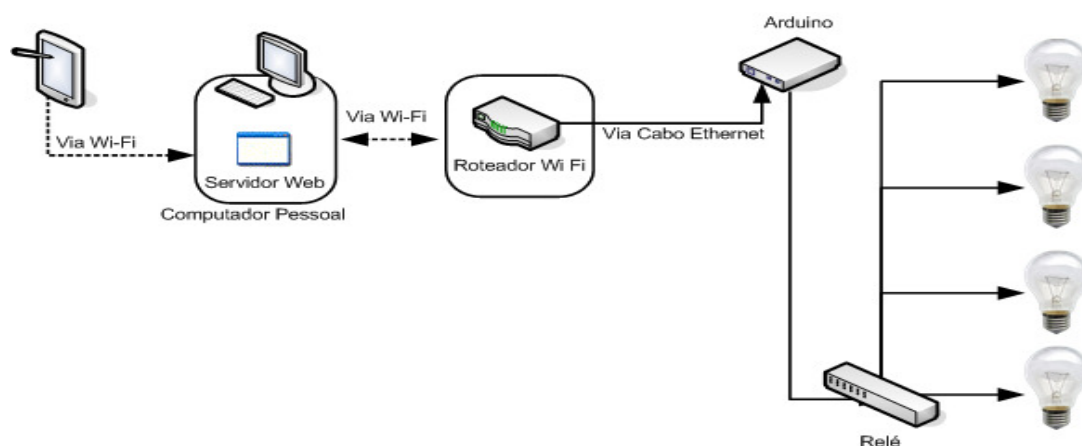


Figura 3.4 – Diagrama Esquemático do projeto proposto

CAPÍTULO 4 – DESCRIÇÃO DO HARDWARE E SOFTWARE

Este capítulo aborda as especificações, de forma detalhada, dos dispositivos utilizados e seu funcionamento nesse projeto, tanto a parte física, ou seja, o *hardware*, como a parte lógica, os *softwares* utilizados.

4.1 Microcontrolador Arduino Uno

Conforme mencionado no capítulo anterior, o microcontrolador utilizado nesse projeto foi o microcontrolador Arduino Uno. O Arduino Uno é uma placa de microcontrolador baseado no ATmega328 . Ele tem 14 pinos de entrada/saída digital, 6 entradas analógicas, um cristal oscilador de 16 MHz, uma conexão USB, uma entrada de alimentação uma conexão ICSP e um botão de reset. Ele contém todos os componentes necessários para suportar o microcontrolador, simplesmente conectando-o a um computador, pela porta USB, ou alimentá-lo com uma fonte/bateria.

4.1.1 Especificações

As principais especificações técnicas do Arduino Uno são:

Tabela 4.1– Especificação do Microcontrolador Arduino Uno.

| | |
|--|--|
| <i>Micro Controlador</i> | <i>ATmega328</i> |
| <i>Voltagem</i> | <i>5V</i> |
| <i>Voltagem de entrada (recomendada)</i> | <i>7-12V</i> |
| <i>Voltagem de entrada (limite)</i> | <i>6-20V</i> |
| <i>Pinos Digitais de I/O</i> | <i>14 (6 são fornecidas pela saída PWM output)</i> |
| <i>Pinos Analógicos</i> | <i>6</i> |
| <i>Corrente por pino de I/O</i> | <i>40 mA</i> |
| <i>Corrente para pino de 3.3V</i> | <i>50 mA</i> |
| <i>Memória Flash</i> | <i>32 KB (ATmega328) desses 0.5 KB são utilizados para carregar o bootloader</i> |
| <i>SRAM</i> | <i>2 KB (ATmega328)</i> |
| <i>EEPROM</i> | <i>1 KB (ATmega328)</i> |
| <i>Velocidade de Clock</i> | <i>16 MHz</i> |

FONTE: <http://arduino.cc/en/Main/ArduinoBoardUno>, disponível em 02/11/2012.

4.1.2 Pinagem do Arduino Uno

Na figura 4.1, é possível ter uma visão da arquitetura do microcontrolador. E seus detalhes foram abordados na tabela 4.1 (localizada no item anterior).

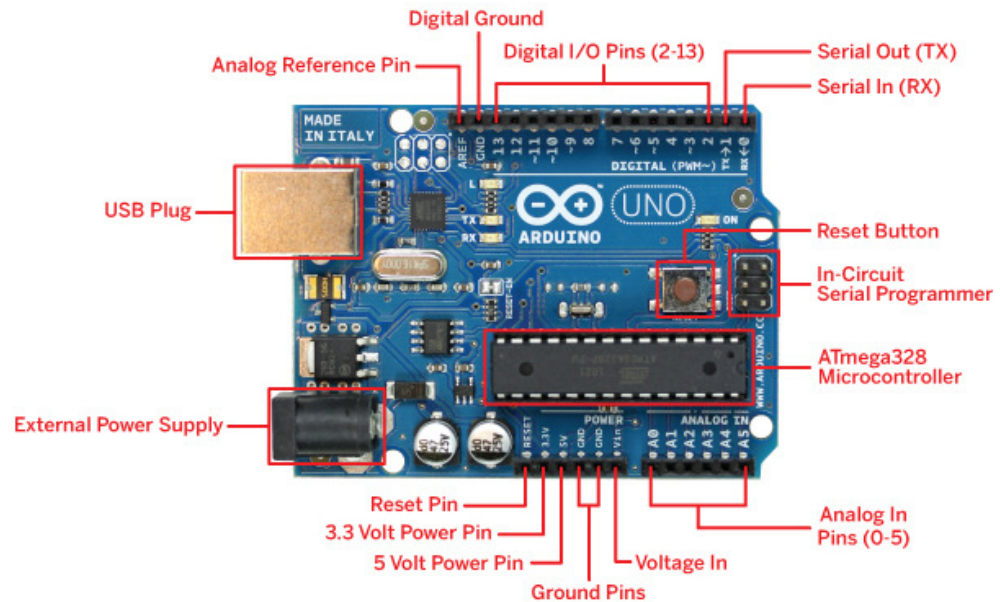


Figura 4.1 – Micro Controlador Arduino Uno
Fonte: Arduino CC,

A pinagem escolhida no projeto pode ser observada na figura 4.2. Essa configuração atende às necessidades de cada componente utilizado no projeto.

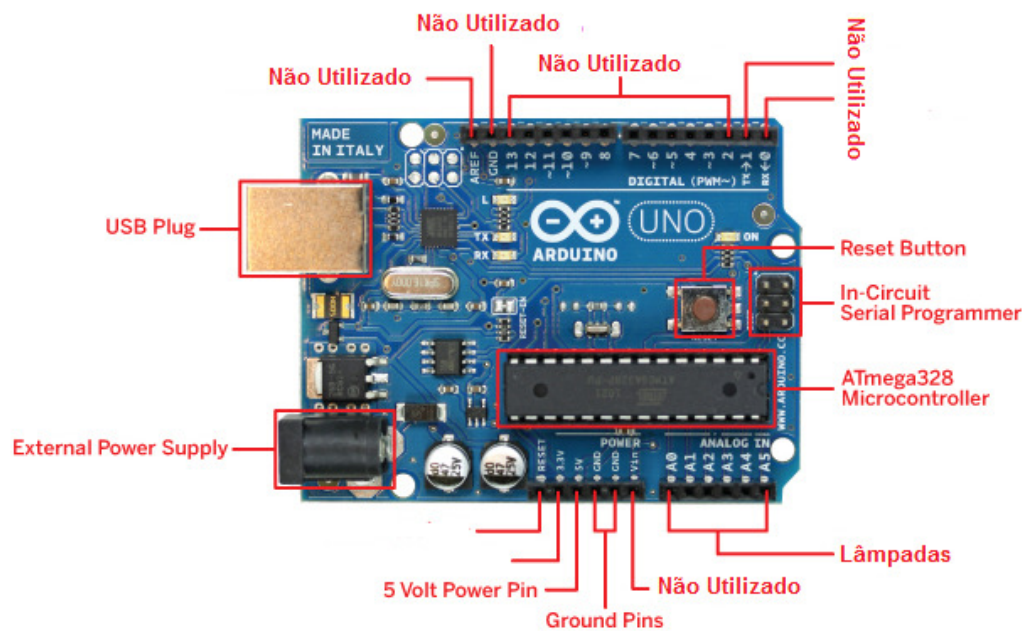


Figura 4.2 – Micro Controlador Arduino
Fonte: Autor

4.2 Relé

Os relés são dispositivos eletromecânicos capazes de controlar circuitos de grandes correntes a partir de pequenas correntes ou tensões. Seu funcionamento é bem simples, quando uma corrente circula pela bobina, essa cria um campo magnético que atrai o contato fechando ou abrindo circuito. Ao cessar a corrente da bobina o campo magnético também cessa e o contato volta a sua posição original.

A placa de interface de relés é utilizada para controlar aparelhos e outros equipamentos com corrente de grande porte, neste projeto esta placa foi utilizada para controlar as lâmpadas do protótipo. Contudo, esta placa ainda pode ser controlada com outros microcontroladores, dentre eles (Arduino , 8051, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic).

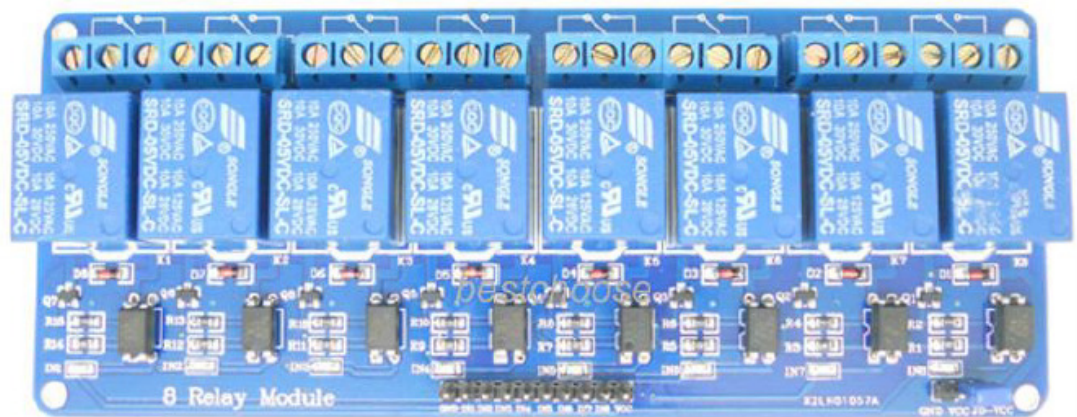


Figura 4.2 – Placa de interface de relés
Fonte: (PHOTOBUCKET, 2011)

4.3 Roteador Linksys WRT54G

O roteador foi responsável por criar a comunicação entre o micro controlador Arduino e o programa que residia no host.

O Linksys possui as seguintes características:

- Cumpre com padrões de 802.11g e 802.11b (2.4GHz)
- Todas as portas LAN são compatíveis com auto-crossover (MDI/MDI-X)

- Compatível com Wired Equivalent Privacy™ (WEP), Wi-Fi Protected Access™ (WPA) e Wi-Fi Protected Access™2 (WPA2)



Figura 4.3 – Roteador Wireless (Fonte: Autor)

O Botão Reset (Repor) repor as predefinições de fábrica do Router. 1, 2, 3, 4 Estas portas (1, 2, 3, 4) ligam o Router aos computadores da rede e a outros dispositivos de rede Ethernet e Power (Alimentação) A porta Power (Alimentação) é onde será ligado o transformador.

4.4 Arduino Software

Para desenvolvimento do código do microcontrolador foi utilizado o *Software Arduino Development Environment*. É um *software* livre e sua linguagem de programação é a *wiring*, que é derivada do C e C ++. Já vem acompanhando por padrão das bibliotecas mais comuns e possibilita a adição de outras já existentes ou a criação de novas. Após a criação código e seus devidos teste, é utilizado a opção de *upload* e dessa forma é carregado o código para o microcontrolador Arduino.

4.5 Programa PHP

Assim como descrito no item 3.5.1(p. 25), a linguagem de programação utilizada foi a PHP. O programa tem como objetivo fazer a comunicação com o microcontrolador arduino e dessa forma facilitar a interação com o usuário. Evitando assim com que esse tenha de possuir conhecimentos técnicos.

CAPÍTULO 5 – IMPLEMENTAÇÃO

Nesse capítulo são apresentados tópicos fundamentais para compreensão geral da implementação do projeto; foram definidas as etapas necessárias para isso, que são:

- Fluxograma Geral do Sistema
- Elaboração do código fonte para o microcontrolador Arduino Uno;
- Montagem dos circuitos nas placas;
- Montagem do protótipo;

Na figura 5.1 é mostrado o protótipo em sua fase final.



Figura 1.1 – Protótipo Concluído (Fonte: Autor)

5.1 – Fluxograma Geral do Sistema

Após definição da disposição dos componentes, foi elaborado o fluxograma geral do sistema, conforme mostra a figura 5.2. Este fluxograma foi fundamental na elaboração do código fonte e na montagem final do protótipo.

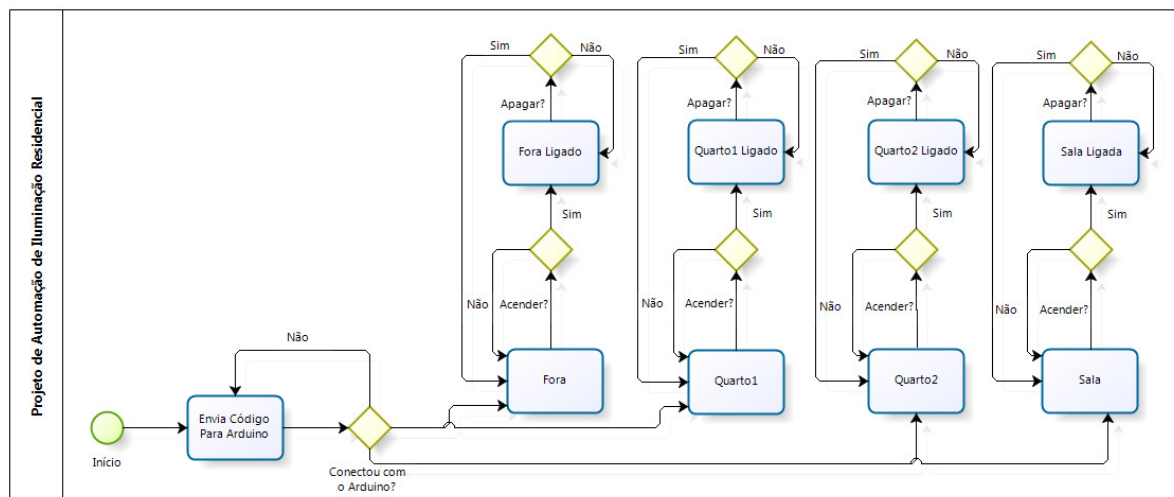


Figura 5.2– Fluxograma Geral do Sistema (Fonte: Autor)

5.2 Elaboração do código fonte

Para realizar a programação do código para o Arduino Uno foi utilizado Arduino Software e, para o servidor web o notepad++.

5.2.1 Para o microcontrolador Arduino Uno

A seguir será explicada a escrita do código fonte, com as funções do programa para o devido funcionamento do projeto com todos seus componentes.


```

#include <SPI.h>
#include <Ethernet.h>
|
//Configurações do Ethernet Shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };// Endereço Mac que o shield ethernet assumirá
byte ip[] = { 192,168,1, 124 }; // ip que o arduino assumirá
byte gateway[] = { 192,168,1, 1 }; // ip do roteador
byte subnet[] = { 255, 255, 255, 0 };

// String que representa o estado dos dispositivos
char Luz[7] = "0000L#";

EthernetServer server = EthernetServer(8081);//EthernetServer server(8081); // Cria o servidor na porta 8081

// String onde é guardada as msgs recebidas
char msg[7] = "0000L#";

```

Figura 5.3– Código Fonte Parte I (Fonte: Autor)

A figura 5.3 mostra o início da escrita do código fonte, em que as bibliotecas necessárias para o funcionamento do programa foram declaradas. A biblioteca SPI foi utilizada, pois ela é consultada pela outra biblioteca chamada Ethernet, a qual concede acesso à internet para a placa do microcontrolador Arduino.

Após a inclusão das bibliotecas que foram utilizadas, o próximo passo foi configurar o endereço MAC, IP, Gateway e Submáscara no *shield ethernet* utilizado. O endereço MAC pode ser qualquer um que já não esteja sendo utilizado na rede local.

Assim que o dispositivo é ativado, ele faz uma leitura do estado das lâmpadas, conforme mencionado na figura 5.2. O dispositivo é sempre ativado com as variáveis setadas como '0', ou seja, desligadas. Para isso, foram criadas duas *strings* com 7 caracteres 'Luz' e 'msg'. A primeira para representar o estado dos dispositivos ao iniciar o dispositivo e a segunda para guardar as mensagens recebidas, informado o comando dado pelo usuário.

Por último, conforme a figura 5.3, configuramos para que seja possível acessar o microcontrolador pela porta 8081.

```

void setup() {
  Ethernet.begin( mac, ip, gateway, subnet);
  server.begin();
  pinMode(A0,OUTPUT);
  pinMode(A1,OUTPUT);
  pinMode(A2,OUTPUT);
  pinMode(A3,OUTPUT);
}

```

Figura 5.4– Código Fonte Parte II (Fonte: Autor)

Nesse ponto foram inicializadas todas as variáveis configuradas anteriormente, conforme mostrado na figura 5.3., por meio da função `Ethernet.begin`. Também foram inicializados os pinos que foram utilizados no código. Uma peculiaridade do microcontrolador arduino uno é que é possível utilizar uma porta analógica como digital. Para isso basta adicionar a letra 'A' antes do número do pino. Foi utilizado dessa forma apenas para facilitar a construção do circuito.

```

void loop() {
  EthernetClient client = server.available();// Client client = server.available();
  // SE receber um caracter...
  if (client) {
    // guarda o caracter na string 'msg'
    msg[1]=msg[2]; msg[2]=msg[3]; msg[3]=msg[4]; msg[4]=msg[5]; msg[5]=msg[6];
    msg[6] = client.read();

    if (msg[6]=='#') {
      switch(msg[5]) {
        case 'R':
          // Se receber o comando 'R#' envia de volta o status dos
          // dispositivos. (Que é a string 'Luz')
          client.write(Luz);
          break;
        case 'L':
          // Caso L#, ele copia os 4 bytes anteriores p/ a
          // string 'Luz' e cada byte representa um
          // dispositivo, onde '1'=ON e '0'=OFF
          Luz[0]=msg[1];
          Luz[1]=msg[2];
          Luz[2]=msg[3];
          Luz[3]=msg[4];
          if (Luz[0]=='1') digitalWrite(A0,HIGH); else digitalWrite(A0,LOW);
          if (Luz[1]=='1') digitalWrite(A1,HIGH); else digitalWrite(A1,LOW);
          if (Luz[2]=='1') digitalWrite(A2,HIGH); else digitalWrite(A2,LOW);
          if (Luz[3]=='1') digitalWrite(A3,HIGH); else digitalWrite(A3,LOW);
          break;

```

Figura 5.5– Código Fonte Parte III (Fonte: Autor)

Nessa parte do código é criado um loop. No primeiro loop é feita uma comparação das variáveis, conforme mostrado na figura 5.3. A variável em questão é a 'msg'. No segundo loop é feito um case para ou verificar o estado das lâmpadas, com a opção 'R' ou enviar um comando de desligar ou ligar, com a opção 'L'. Esse valor é setado pelo programa web no microcontrolador e em seguida é lido pelo programa web.

Caso a variável com a posição de número 5 na string 'msg' seja 'L' é setado o valor da string 'Luz' na string 'msg'.

5.2.2 Para o servidor web

O servidor web será o responsável por enviar o comando dado pelo usuário para o microcontrolador e assim mudar o valor das suas variáveis alterando o estado das lâmpadas. O programa foi criado com a linguagem PHP e é dividido em quatro partes, conexão com o microcontrolador arduino, executando a ação enviada pelo usuário, lendo o estado das lâmpadas e uma saída informando caso não seja possível conectar o programa ao microcontrolador.

```

1  <html>
2  <head></head>
3  <body>
4
5  <?php
6  $sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
7  // Se conecta ao IP e Porta:
8  socket_connect($sock, "192.168.1.124", 8081);
9

```

Figura 5.6– Código Fonte Parte IV (Fonte: Autor)

Conforme figura 5.6, é iniciado o programa e configurada a forma de conexão com o microcontrolador.

```

10 // Executa a ação correspondente ao botão apertado.
11 if(isset($_POST['bits'])) {
12     $msg = $_POST['bits'];
13     if(isset($_POST['Fora'])) { if($msg[0]=='1') { $msg[0]='0'; } else { $msg[0]='1'; }}
14     if(isset($_POST['Quarto1'])) { if($msg[1]=='1') { $msg[1]='0'; } else { $msg[1]='1'; }}
15     if(isset($_POST['Quarto2'])) { if($msg[2]=='1') { $msg[2]='0'; } else { $msg[2]='1'; }}
16     if(isset($_POST['Sala'])) { if($msg[3]=='1') { $msg[3]='0'; } else { $msg[3]='1'; }}
17     socket_write($sock,$msg,strlen($msg));
18 }
19
20 socket_write($sock,'R#',2); //Requisita o status do sistema.

```

Figura 5.7– Código Fonte Parte V (Fonte: Autor)

Nesse ponto do programa, é feita a comunicação direta do programa web com o microcontrolador Arduino através do primeiro loop, alterando o valor da variável msg, mencionada no item 5.2.1(p. 33) e, também é feita a verificação do estado das lâmpadas conforme a linha de número 20.

```

21
22 // Espera e lê o status e define a cor dos botões de acordo.
23 $status = socket_read($sock,6);
24 if (($status[4]=='L') && ($status[5]=='#')) {
25     if ($status[0]=='0') $cor1 = 'lightcoral';
26     else $cor1 = 'lightgreen';
27     if ($status[1]=='0') $cor2 = 'lightcoral';
28     else $cor2 = 'lightgreen';
29     if ($status[2]=='0') $cor3 = 'lightcoral';
30     else $cor3 = 'lightgreen';
31     if ($status[3]=='0') $cor4 = 'lightcoral';
32     else $cor4 = 'lightgreen';
33

```

Figura 5.8– Código Fonte Parte VI (Fonte: Autor)

Para o início da terceira parte do código, que tem como foco configurar a cor dos botões do programa, verde como ligado e vermelho como desligado. Em seguida são exibidos os botões com seu tamanho e fontes configurados, conforme a figura 5.9 logo abaixo.

```

34     echo "<form method = \"post\" action = \"PaginaPHP.php\">";
35     echo "<input type = \"hidden\" name = \"bits\" value = \"\$status\">";
36     echo "<button style = \"width:950; background-color: $cor1 ;font: bold 120px Arial\" type = \"Submit\" Name = \"Fora\">Fora</button><br><br>";
37     echo "<button style = \"width:950; background-color: $cor2 ;font: bold 120px Arial\" type = \"Submit\" Name = \"Quarto1\">Quarto1</button><br><br>";
38     echo "<button style = \"width:950; background-color: $cor3 ;font: bold 120px Arial\" type = \"Submit\" Name = \"Quarto2\">Quarto2</button><br><br>";
39     echo "<button style = \"width:950; background-color: $cor4 ;font: bold 120px Arial\" type = \"Submit\" Name = \"Sala\">Sala</button></br></br>";
40
41     echo "</form>";
42 }
43 // Caso ele não receba o status corretamente, avisa erro.
44 else { echo "Falha ao receber status da casa." ; }
45 socket_close($sock);
46 ?>
47
48 </body>
49 </html>

```

Figura 5.8– Código Fonte Parte VI (Fonte: Autor)

Na quarta e última parte do código foi feita a verificação da conexão entre o programa e o microcontrolador. Caso a conexão não seja feita é exibida uma mensagem de erro informando que não foi possível receber o estado das lâmpadas.

5.3 Elaboração do circuito

Após determinar o que terá e como será o funcionamento do projeto foi criado desenvolvido o circuito e desenvolvida a placa de relé.

5.3.1 Placa Relé

Para o desenvolvimento da placa de relé foi utilizado o *software* ARES 7.7 Professional. Com tal *software* é possível projetar o circuito e as trilhas. Na figura 5.9, segue a placa projetada.

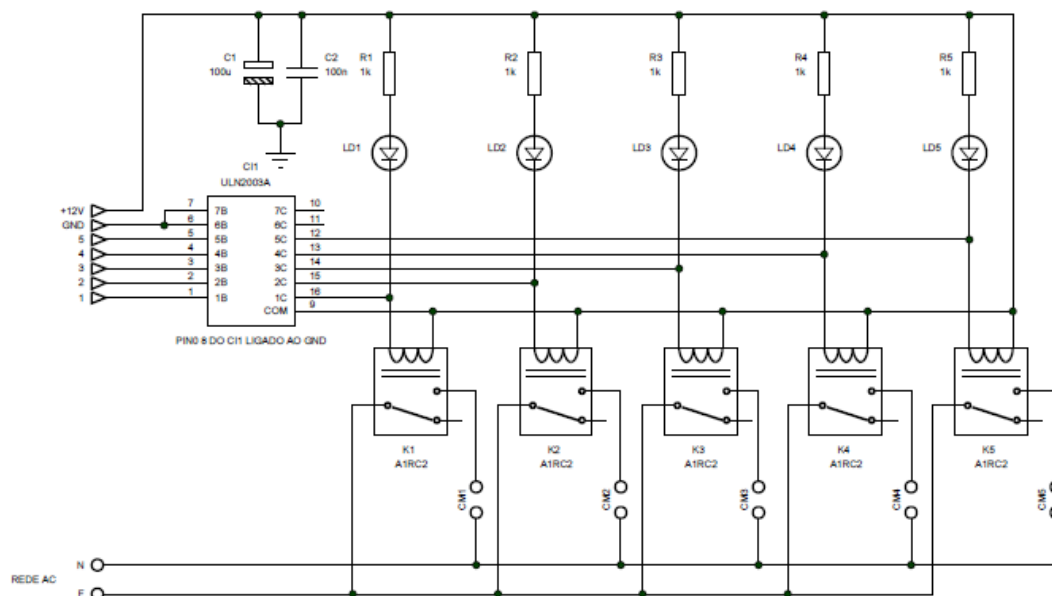


Figura 5.9– Desenvolvimento Placa Relé (Fonte: Autor)

5.3.2 Montagem do Circuito

O circuito, conforme figura 5.10 a seguir, é utilizado para fazer a ligação entre a placa de relés e o microcontrolador Arduino. Foram utilizadas cinco pinos, dentre eles o terra (GND – *Graduated Neutral Density Filter*) e os outros quatro teoricamente analógicos. Teoricamente devido a uma característica do microcontrolador arduino que quando adicionamos a letra ‘A’ junto a um pino analógico ele se torna digital. Foi optado por utilizá-los devido à proximidade com o pino GND.

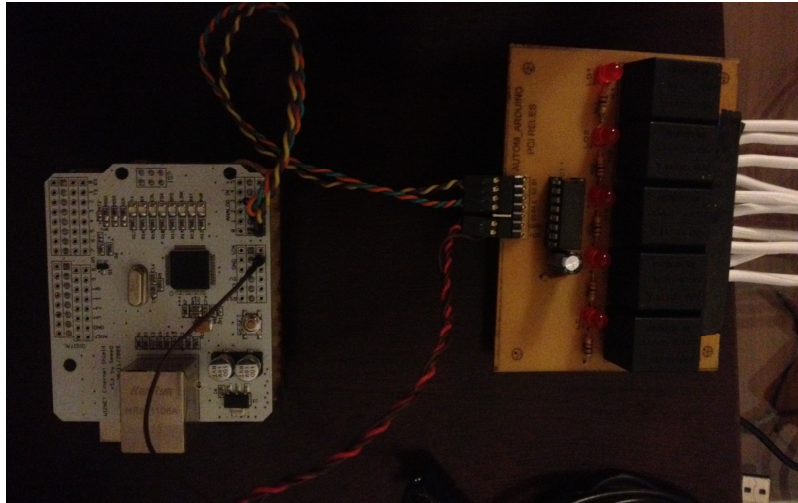


Figura 5.10– Montagem Circuito (Fonte: Autor)

CAPÍTULO 6 – RESULTADOS OBTIDOS

Neste capítulo são apresentados os resultados obtidos para chegar à solução do problema apresentado no Capítulo 2 – Apresentação do Problema – através da realização de simulações. São apresentados também alguns problemas encontrados.

6.1 Simulações

As primeiras simulações não foram feitas com lâmpadas. Os primeiros testes realizados foram feitos com led de alto brilho e sem a placa de relé envolvida no circuito.

6.2 Problemas Encontrados

O primeiro problema inesperado foi que o programa utilizado para hospedar o servidor web não ter configurado uma biblioteca para leitura de *socket*. O programa compilava sem erros, porém na hora de sua execução tinha como erro não conseguir ler o estado dos componentes.

O segundo erro foi que o servidor web não reconhecia as cores programadas para o estado dos botões. O erro foi resolvido apenas com a definição correta para as variáveis que hospedavam essa cor.

O terceiro e mais complicado problema ocorreu com o *shield ethernet*. Por algum motivo após a realização de alguns testes apenas uma lâmpada continuou a funcionar. Foi adquirido outro *shield* com a mesma configuração e assim o problema foi resolvido.

6.3 Orçamento do Projeto

A aquisição dos componentes eletrônicos para o projeto foi realizada e foi fundamental para o projeto. A tabela 6.1 mostra os custos de cada componente.

Tabela 6.1– Orçamento Projeto

| Item | Custo Unitário | Quantidade | Custo Total |
|-------------------------------------|-----------------------|-------------------|--------------------|
| Microcontrolador Arduino Uno | R\$ 78,00 | 1 | R\$ 78,00 |
| Arduino Ethernet Shield | R\$ 90,00 | 1 | R\$ 90,00 |
| Iphone 4s | R\$ 1.299,00 | 1 | R\$ 1.299,00 |
| Placa Relé | R\$ 40,00 | 1 | R\$ 100,00 |
| Lâmpada | R\$ 4,00 | 6 | R\$ 24,00 |
| Tomada Macho | R\$ 0,30 | 6 | R\$ 1,80 |
| Papel | R\$ 12,50 | 2 | R\$ 25,00 |
| Madeira | R\$ 10,50 | 1 | R\$ 10,50 |
| Total | | | R\$ 1.628,30 |

FONTE: Autor

CAPÍTULO 7 – CONSIDERAÇÕES FINAIS

7.1 Conclusões

Foi desenvolvido neste trabalho um protótipo de uma residência que, por meio de um dispositivo Iphone/Ipad, fosse possível controlar a iluminação de tal lugar.

O projeto funcionou conforme o previsto, e desta forma pode ser usado para o seu devido fim, automação de iluminação residencial via Ipad/Iphone.

O objetivo do projeto foi alcançado e o protótipo funcionou conforme o planejado. Ocorreram alguns imprevistos durante o processo de confecção, mas tais incidentes, comentados na seção 6.2, foram resolvidos.

7.2 Propostas para Trabalhos Futuros

Uma sugestão para trabalhos futuros seria a de adicionar controle de portas e portões, fazer a ligação também com interruptores para o usuário conseguir controlar a iluminação de sua casa tanto via Ipad/Iphone quanto por interruptores.

A segunda sugestão é de configurar uma página que solicite usuário e senha conseguindo assim uma maior segurança quanto ao controle da iluminação de sua residência.

REFERÊNCIAS BIBLIOGRÁFICAS

LEAL, R. Viva na casa do futuro hoje. **Info Notícias**. Publicado em: 17 jan. 2011. Disponível em: <<http://info.abril.com.br/noticias/tecnologia-pessoal/viva-na-casa-do-futuro-hoje-17012011-2.shl>> Acesso em 29 mar. 2011.

MURATORI, J. R. **Estamos preparados para a automação residencial?** Publicado em: [2010]. Disponível em: <<http://www.forumdaconstrucao.com.br/conteudo.php?a=11&Cod=825>> Acesso em 30 mar. 2011.

PHOTOBUCKET, 2011 PHOTOBUCKET. Disponível em: <http://i1134.photobucket.com/albums/m609/onlyforever702/CNC/8ch5v.jpg>. Acesso em: 20/10/2011.

TEZA, V. R. **Alguns aspectos sobre a Automação Residencial - Domótica.** Dissertação submetida à Universidade Federal de Santa Catarina, 2002.

TANENBAUM, A. S. **Redes de Computadores**, 1997. P 617.

APÊNDICE A - CÓDIGO FONTE DO DISPOSITIVO

```
#include <SPI.h>

#include <Ethernet.h>

//Configurações do Ethernet Shield

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

byte ip[] = { 192,168,1, 124 }; // ip que o arduino assumirá

byte gateway[] = { 192,168,1, 1 }; // ip do roteador

byte subnet[] = { 255, 255, 255, 0 };

// String que representa o estado dos dispositivos

char Luz[7] = "0000L#";

EthernetServer server = EthernetServer(8081); // EthernetServer server(8081); // Cria o servidor na
porta 8081

// String onde é guardada as msgs recebidas

char msg[7] = "0000L#";

void setup() {

  Ethernet.begin( mac, ip, gateway, subnet);

  server.begin();

  pinMode(A0,OUTPUT);

  pinMode(A1,OUTPUT);

  pinMode(A2,OUTPUT);

  pinMode(A3,OUTPUT);

}
```

```

void loop() {

EthernetClient client = server.available();// Client client = server.available();

// SE receber um caracter...

if (client) {

// guarda o caracter na string 'msg'

msg[1]=msg[2]; msg[2]=msg[3]; msg[3]=msg[4]; msg[4]=msg[5]; msg[5]=msg[6];

msg[6] = client.read();

if (msg[6]=='#') {

switch(msg[5]) {

case 'R':

// Se receber o comando 'R#' envia de volta o status dos

// dispositivos. (Que é a string 'Luz')

client.write(Luz);

break;

case 'L':

// Caso L#, ele copia os 4 bytes anteriores p/ a

// string 'Luz' e cada byte representa um

// dispositivo, onde '1'=ON e '0'=OFF

Luz[0]=msg[1];

Luz[1]=msg[2];

Luz[2]=msg[3];

Luz[3]=msg[4];

if (Luz[0]=='1') digitalWrite(A0,HIGH); else digitalWrite(A0,LOW);

if (Luz[1]=='1') digitalWrite(A1,HIGH); else digitalWrite(A1,LOW);

if (Luz[2]=='1') digitalWrite(A2,HIGH); else digitalWrite(A2,LOW);

if (Luz[3]=='1') digitalWrite(A3,HIGH); else digitalWrite(A3,LOW);

```

```
break;
```

```
}
```

```
}
```

```
}
```

```
}
```

APÊNDICE B - CÓDIGO FONTE WEB

```

<html>

<head></head>

<body>

<?php

$sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);

// Se conecta ao IP e Porta:

socket_connect($sock,"192.168.1.124", 8081);

// Executa a ação correspondente ao botão apertado.

if(isset($_POST['bits'])) {

    $msg = $_POST['bits'];

    if(isset($_POST['Fora' ])){ if($msg[0]=='1') { $msg[0]='0'; } else { $msg[0]='1'; }}

    if(isset($_POST['Quarto1'])){ if($msg[1]=='1') { $msg[1]='0'; } else { $msg[1]='1'; }}

    if(isset($_POST['Quarto2'])){ if($msg[2]=='1') { $msg[2]='0'; } else { $msg[2]='1'; }}

    if(isset($_POST['Sala' ])){ if($msg[3]=='1') { $msg[3]='0'; } else { $msg[3]='1'; }}

    socket_write($sock,$msg,strlen($msg));

}

socket_write($sock,'R#',2); //Requisita o status do sistema.

// Espera e lê o status e define a cor dos botões de acordo.

$status = socket_read($sock,6);

if (($status[4]=='L')&&($status[5]=='#')) {

    if ($status[0]=='0') $cor1 = 'lightcoral';

    else $cor1 = 'lightgreen';

```

```

if ($status[1]==0) $cor2 = 'lightcoral';
    else $cor2 = 'lightgreen';
if ($status[2]==0) $cor3 = 'lightcoral';
    else $cor3 = 'lightgreen';
if ($status[3]==0) $cor4 = 'lightcoral';
    else $cor4 = 'lightgreen';

echo "<form method =\"post\" action=\"PaginaPHP.php\">";
echo "<input type=\"hidden\" name=\"bits\" value=\"${status}\">";

echo "<button style=\"width:950; background-color: $cor1 ;font: bold 120px Arial\" type = \"Submit\"
Name = \"Fora\">Fora</button></br></br>";

echo "<button style=\"width:950; background-color: $cor2 ;font: bold 120px Arial\" type = \"Submit\"
Name = \"Quarto1\">Quarto1</button></br></br>";

echo "<button style=\"width:950; background-color: $cor3 ;font: bold 120px Arial\" type = \"Submit\"
Name = \"Quarto2\">Quarto2</button></br></br>";

echo "<button style=\"width:950; background-color: $cor4 ;font: bold 120px Arial\" type = \"Submit\"
Name = \"Sala\">Sala</button></br></br></br>";

echo "</form>";
}

// Caso ele não receba o status corretamente, avisa erro.
else { echo "Falha ao receber status da casa."; }

socket_close($sock);

?>

</body>
</html>

```