

ANDRÉ OSTI SAKAGUCHI

SISTEMA DE MONITORAMENTO E CONTROLE PARA RESIDÊNCIAS COM USO
DE SOFTWARE LIVRE



CURITIBA
2014



ANDRÉ OSTI SAKAGUCHI

SISTEMA DE MONITORAMENTO E CONTROLE PARA RESIDÊNCIAS COM USO SOFTWARE LIVRE

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina TE105 - Projeto de Graduação, do Curso Superior de Engenharia Elétrica, do Departamento Acadêmico de Engenharia Elétrica, da Universidade Federal do Paraná – UFPR, como requisito para obtenção do título de Engenheiro Eletricista.

Orientador:
Prof. Dr. James Alexandre Baraniuk

TERMO DE APROVAÇÃO

ANDRÉ OSTI SAKAGUCHI

SISTEMA DE MONITORAMENTO E CONTROLE PARA RESIDÊNCIAS COM USO DE SOFTWARE LIVRE

Trabalho de conclusão de Curso de Graduação aprovado como requisito parcial para obtenção do grau de Engenheiros Eletricistas no Curso de Engenharia Elétrica, Setor de Tecnologia da Universidade Federal do Paraná, pela seguinte banca examinadora:

Prof. Dr. James Alexandre Baraniuk
Orientador - Departamento Acadêmico de Engenharia Elétrica
da Universidade Federal do Paraná, UFPR.

Prof. Dr. Eduardo Parente Ribeiro
Departamento Acadêmico de Engenharia Elétrica
da Universidade Federal do Paraná, UFPR.

Prof. Dr. João Américo Vilela Junior
Departamento Acadêmico de Engenharia Elétrica
da Universidade Federal do Paraná, UFPR.

Curitiba, 10 de Junho de 2014.

DEDICATÓRIA

Eu, André, dedico esse trabalho de conclusão de curso àqueles que são indispensáveis na minha vida: meu pai Sergio, minha mãe Waldizia e minha namorada Talita.

AGRADECIMENTOS

Agradeço a todas as pessoas que me acompanharam para realização deste trabalho durante o curso de Engenharia Elétrica.

Meu especial agradecimento a meu orientador Prof. Dr. James Alexandre Baraniuk, por se esforçar em passar todo seu conhecimento e se empenhar em me ajudar para realização deste trabalho de conclusão de curso.

Agradeço à Deus, por me dar forças e fazer com que eu realize mais um sonho em minha vida: me tornar engenheiro.

Aos meus pais, Sergio Taidi Sakaguchi e Waldizia Marques Osti Sakaguchi, por todo o apoio, carinho e amor que recebi fazendo me tornar uma pessoa melhor. À minha irmã, Patrícia Osti Sakaguchi, pela paciência e apoio nesses anos de graduação.

À minha namorada Talita Aguiar Coelho pelo carinho, apoio, incentivo e amor dado nas horas mais difíceis e que sem ela não conseguiria seguir em frente.

"No meio da dificuldade encontra-se a oportunidade."

Albert Einstein

RESUMO

Automação residencial é uma das tecnologias promissoras que cresceram consideravelmente no mundo. Qualidade de vida, conforto, segurança são alguns dos benefícios que as pessoas podem ter após se mudarem para uma casa inteligente. Com um sistema inteligente e programável, os equipamentos de um domicílio passam a ser controlados e supervisionados local e/ou remotamente de forma automática. Este trabalho foca no desenvolvimento e construção de um sistema de controle e monitoramento de residências com uso de hardware livre e de diferentes tipos de comunicação entre os componentes deste sistema, tais como: RS485, rádio frequência de 433MHz e 2,4GHz, bluetooth, infravermelho e ethernet. Através de um supervisor o usuário poderá supervisionar e comandar o sistema por completo.

Palavras Chave: Automação residencial, Sistema, Comunicação, Supervisor.

ABSTRACT

Home automation is one of the promising technologies that have been growing considerably in the world. Quality of life, comfort, security are some of the benefits that people can have after moving to a smart home. With an intelligent and programmable system, the equipment of a home shall be controlled and monitored locally or remotely in an automatically way.

This work focuses on the development and construction of a system of control and monitoring of houses using open hardware and different types of communication between the system components, such as: RS485, radio frequency of 433MHz and 2.4GHz, Bluetooth, infrared and ethernet. Through a supervisory the user can make the total supervision and control of the system.

Keywords: Home Automation, System, Communication, Supervision.

LISTA DE ILUSTRAÇÕES

FIGURA 1– ILUSTRAÇÃO DAS VARIÁVEIS A SEREM CONTROLADAS	13
FIGURA 2 - ESQUEMÁTICO DO SISTEMA	1
FIGURA 3 - ESQUEMÁTICO USO DE RELÉ	16
FIGURA 4 - RELÉ	17
FIGURA 5 - ESQUEMÁTICO MÓDULO RELÉ	18
FIGURA 6 - SENSOR DE PRESENÇA PIR	19
FIGURA 7 - SENSOR DE UMIDADE	21
FIGURA 8 - ESTRUTURA DE UM MICROCONTROLADOR.....	22
FIGURA 9 - KIT DE DESENVOLVIMENTO ARDUINO UNO	23
FIGURA 10 - ATMEGA328-PU	23
FIGURA 11 - PINAGEM ATMEGA328-PU	24
FIGURA 12 - PINAGEM ATTINY85	24
FIGURA 13 - MAX485.....	27
FIGURA 14 - RECEPTOR E TRANSMISSOR RF433MHZ.....	30
FIGURA 15 - NRF24L01	31
FIGURA 16 - MÓDULO ETHERNET.....	34
FIGURA 17 - SENSOR IR	35
FIGURA 18 - CONTROLE DE ACESSO	37
FIGURA 19 - PÁGINA DA CASA INTELIGENTE	38
FIGURA 20 - PÁGINA REFERENTE AO AMBIENTE 1	39
FIGURA 21 – ESQUEMÁTICO 1	40
FIGURA 22 – ESQUEMÁTICO 2	41
FIGURA 23 – ESQUEMÁTICO 3	41
FIGURA 24 – ESQUEMÁTICO 4	42
FIGURA 25 – CENTRAL	48
FIGURA 26 – MÓDULO	48
FIGURA 27 – PCB MÓDULO ATMEGA328.....	50
FIGURA 28 – PLACA MÓDULO ATMEGA328	50
FIGURA 29 – ESQUEMÁTICO MÓDULO ATTINY85	51
FIGURA 30 – PCB MÓDULO ATTINY85.....	52
FIGURA 31 – PLACA MÓDULO ATTINY85.....	52
FIGURA 32 – ESQUEMÁTICO MÓDULO ATMEGA328.....	85
FIGURA 33 – AMBIENTE DE TRABALHO PARA O APLICATIVO.....	87
FIGURA 34 – PROGRAMAÇÃO PARA O APLICATIVO.....	88
FIGURA 35 – PROGRAMAÇÃO CONEXÃO BLUETOOTH.....	88
FIGURA 36 – PROGRAMAÇÃO ACIONAMENTO POR BOTÕES.....	89

LISTA DE TABELAS

TABELA 1 – CÓDIGOS ENVIADOS PELO CONTROLE	36
TABELA 2 – CÓDIGO DO CONTROLE E SUA FUNÇÃO NO SISTEMA	36
TABELA 3 – BIBLIOTECAS	42
TABELA 4 – CÓDIGO ENVIADO PARA OS MÓDULOS	46
TABELA 5 – SIMBOLOGIA CENTRAL	48
TABELA 6 – SIMBOLOGIA MÓDULO	49
TABELA 7 – LISTA DE ITENS UTILIZADOS	54

LISTA DAS SIGLAS

ABNT	-	Associação Brasileira de Normas Técnicas
AM	-	Modulação em Amplitude
ASK	-	<i>Amplitude Shift Keying</i>
AURISIDE	-	Associação Brasileira de Automação Residencial
Byte	-	Unidade de armazenamento
CPU	-	Unidade Central de Processamento
<i>Datasheet</i>	-	Folha de dados
DHT22	-	Sensor de umidade
EEPROM	-	<i>Electrically-Erasable Programmable Read-Only Memory</i>
EIA	-	<i>Electronic Industries Association</i>
IBGE	-	Instituto Brasileiro de Geografia e Estatística
ICSP	-	<i>In Circuit Serial Programming</i>
IDE	-	<i>Integrated Development Environment</i>
IP	-	<i>Internet Protocol</i>
KB	-	<i>Kilo Byte</i>
LM35	-	Sensor de temperatura
MDF	-	<i>Medium Density Fiberboard</i>
MCU	-	<i>Micro Controller Unit</i>
nRF24L01	-	Módulo de rádio frequência de 2,4GHz
PCB	-	<i>Printed Circuit Board</i>
PHP	-	Personal Home Page
PWM	-	<i>Pulse width modulation</i>
RS	-	<i>Recommended Standard</i>
SRAM	-	<i>Static Random Access Memory</i>
TCP	-	<i>Transmission Control Protocol</i>
TV	-	televisão
UDP	-	<i>User Datagram Protocol</i>
Vcc	-	Tensão contínua
Vca	-	Tensão alternada
Web	-	<i>World Wide Web</i>

SUMÁRIO

RESUMO.....	VII
ABSTRACT	VIII
LISTA DE ILUSTRAÇÕES	1
LISTA DE TABELAS	3
LISTA DAS SIGLAS	4
SUMÁRIO.....	5
1.1 CONTEXTUALIZAÇÃO	7
1.2 OBJETIVOS	8
1.3 JUSTIFICATIVA	8
1.4 DIFERENCIAL EM RELAÇÃO A OUTROS PROJETOS	9
1.5 ESTRUTURA DA MONOGRAFIA	10
2 AUTOMAÇÃO RESIDENCIAL	11
3 IMPLEMENTAÇÃO DO PROJETO	14
3.1 EQUIPAMENTOS	15
3.1.1 Atuadores	16
3.1.2 Sensores	18
3.2 MICROCONTROLADOR.....	21
3.2.1 ATmega328.....	23
3.2.2 ATtiny85	24
3.3 FONTE DE ALIMENTAÇÃO.....	25
3.4 CENTRAL.....	25
3.5 MÓDULO.....	26
3.6 COMUNICAÇÃO	27
3.6.1 RS485	27
3.6.2 RÁDIO FREQUENCIA.....	29
3.6.3 Bluetooth	32
3.6.4 Ethernet.....	33
3.6.5 INFRATERMELHO	35
3.6.6 SUPERVISÓRIO	37
3.7.1 Introdução	39
3.7.2 Programação do supervisor web	43

3.7.3 Programação da central	43
3.7.4 Programação dos módulos.....	46
4 PROTÓTIPO	47
4.1 PRIMEIRA ETAPA	47
4.2 SEGUNDA ETAPA.....	49
4.3 TERCEIRA ETAPA	53
5 CONCLUSÃO.....	55
5.1 CONSIDERAÇÕES FINAIS	55
5.2 TRABALHOS FUTUROS	55
REFERÊNCIAS.....	57
APÊNDICE A – PÁGINA HOME	61
APÊNDICE B – PÁGINA AMBIENTE 1	64
APÊNDICE C – PROGRAMAÇÃO CENTRAL	68
APÊNDICE D – PROGRAMAÇÃO MÓDULO 1	78
APÊNDICE E – PROGRAMAÇÃO MÓDULO 2	82
APÊNDICE F – ESQUEMÁTICO MÓDULO 1.....	85
APÊNDICE G – PCB MÓDULOS.....	86
APÊNDICE H – APLICATIVO ANDROID.....	87
ANEXO A – DHT22	90
ANEXO B – MAX485.....	91
ANEXO C – nrf24l01	92

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A evolução da tecnologia sempre esteve pautada no atendimento das novas necessidades do homem que este está sempre em busca do desenvolvimento de novos mecanismos que possam lhe trazer benefícios como conforto, qualidade de vida, segurança com custos decrescentes. A revolução da maneira como as pessoas se conectam e se comunicam hoje em dia somado a popularização dos equipamentos eletrônicos proporciona uma vasta gama de praticidade na realização de tarefas do cotidiano jamais antes pensadas.

A automação residencial surgiu justamente para melhorar a qualidade de vida, segurança, praticidade, conforto e se executa de maneira eficiente resultando em economia de energia dentro de uma residência.

Segundo Bolzani (2004), “a automação residencial vem com o objetivo de integrar equipamentos e serviços de maneira em que eles fiquem centralizados em um sistema inteligente e programável, através do qual sejam possíveis o controle e supervisão de diversas tarefas de modo automático”. A preocupação crescente com a sustentabilidade resultando na busca por processos cada vez mais eficientes para redução de consumo de energia está cada vez mais presente na realidade diária de empresas, de organizações não governamentais e da sociedade. Um dos estudos em relação a esse assunto é a implementação das redes inteligentes de energia *Smart Grid*. O uso deste conceito aliado com uma casa inteligente resulta em uma das vantagens que esta proporciona aos moradores.

Segundo dados de mercado levantados pela Associação Brasileira de Automação Residencial (AURESIDE, 2014), ocorreu entre os anos de 2008 e 2011 o crescimento de 120% de aumento de empresas com interesse em oferecer soluções de automação residencial no Brasil. Este aumento configurou uma queda de 50% nos valores finais dos produtos e serviços ofertados nos últimos cinco anos. Este panorama favorável aliado a soluções de fácil instalação faz com que o volume de projetos com automação no Brasil cresça de 30 a 40% ao ano e ainda deve continuar crescendo neste ritmo por um longo período, pois o mercado ainda é pouco explorado.

1.2 OBJETIVOS

O principal objetivo deste trabalho é criar e tornar operacional um sistema para automatizar uma residência utilizando diferentes tipos de comunicação. Para a comunicação entre o usuário e a central de controle serão utilizados os seguintes métodos: *ethernet*, *bluetooth* e infravermelho. Já a comunicação entre o controlador e os módulos de automação utilizados em cada ambiente da residência será feita através de RS485, Radio Frequência de 433Mhz e 2,4Ghz.

Para o alcance do objetivo principal foram determinados objetivos secundários que são: contextualização da automação residencial no cenário brasileiro; composição de um sistema de automação residencial; estudo de diferentes tipos de comunicação; implementação de um protótipo.

1.3 JUSTIFICATIVA

Segundo a CISCO (2013), empresa líder mundial na transformação em como as pessoas se conectam, se comunicam e colaboram, a "internet das coisas" é apontada como a próxima onda da tecnologia. A ideia de disponibilizar conexão em todos os lugares e fazer com que os objetos "conversem" é a grande aposta da Cisco, segundo a qual este será um mercado de US\$ 19 trilhões na próxima década.

A popularização dos dispositivos de automação que podem ser conectados em rede permitindo o controle de diversas variáveis de interesse remotamente, vem ao mesmo tempo tornando este tipo de tecnologia mais acessível ao público e abrindo novos caminhos de possibilidades para esta nova era.

Uma das aplicações deste conceito que tem mostrado uma rápida evolução e aceitação com certeza é a automação residencial. Casas inteligentes controladas por aplicativos de celular, ou via internet, com capacidade de trocar informações entre equipamentos eletrônicos e que permitem a interação do usuário remotamente estão cada vez mais presentes na vida das pessoas, garantindo muito mais conforto, segurança e economia a este tipo de ambiente.

Segundo dados retirados do censo demográfico de 2010 do IBGE, no Brasil existem 57.324.167 domicílios particulares permanentes onde 56.595.495 possuem energia elétrica. De acordo com a Associação Brasileira de Automação Residencial (AURESIDE) cerca de 300.000 casas são automatizadas no Brasil e a quantidade de interessados por esta tecnologia vem crescendo consideravelmente.

Tendo em vista o status da evolução desta tecnologia aliado às vantagens acima apresentadas e ao cenário brasileiro, que está se tornando favorável, decidiu-se abordar mais a fundo características técnicas sobre a automação de residências neste trabalho de conclusão de curso.

1.4 DIFERENCIAL EM RELAÇÃO A OUTROS PROJETOS

Verificou-se os trabalhos relacionados ao tema que foram realizados no curso de Engenharia Elétrica da Universidade Federal do Paraná, sendo apresentado na sequência os que se relacionam diretamente ao tema.

Domótica voltada para plataformas móveis com sistema operacional *Android*. Neste trabalho o objetivo foi de desenvolver um aplicativo para plataformas móveis para realizar o gerenciamento e utilizar o microcontrolador LPC1768 para realizar a automação dos dispositivos (MARUAN B. MENDES e THIAGO M. MARCENGO, 2011).

Sistema embarcado com acesso sem fio. Neste projeto foi utilizado um *hardware* genérico de roteamento sem fio, um equipamento multifuncional que atende algumas necessidades de um escritório, como por exemplo: Serviços de compartilhamento de arquivos e impressoras, rede sem fio, monitoramento com câmera, roteamento e armazenamento de arquivos (EDUARDO M. NICKEL e WILLIAM K. S. M. BESSA, 2010).

Automação residencial utilizando protocolo CAN. O protocolo CAN teve o principal enfoque do trabalho, porém foi feito um resumo dos demais protocolos existentes em automação residencial. Foi criado um sistema com o protocolo CAN utilizando um computador pessoal como integrador da rede de dados CAN, os componentes usados e um *software* para integrar todos os equipamentos (EDUARDO H. C. FERREIRA, 2009).

Domótica automação residencial. Trabalho realizado com ênfase em explicar o que é a automação residencial e os padrões utilizados pelos fabricantes (SUHELEN R. HERNASKI, 2010).

O trabalho que será realizado a seguir se diferencia dos demais citados acima com especial enfoque em como o sistema se integrará com os equipamentos. Portanto o foco será nas comunicações que irão ocorrer entre todos os equipamentos do sistema. Existirão diferentes tipos de comunicação tais como: *ethernet*, *bluetooth*, infra-vermelho, rádio frequência e RS485. Estes variados tipos de comunicação darão flexibilidade ao sistema de automação residencial.

1.5 ESTRUTURA DA MONOGRAFIA

Este trabalho de conclusão de curso está dividido em cinco capítulos. O primeiro capítulo apresenta a contextualização do tema abordado, os objetivos principais e secundários, a justificativa da importância do tema assim como o diferencial em relação a outros trabalhos de conclusão de curso. O capítulo 2 revisa os conceitos sobre automação residencial. No terceiro capítulo é feita uma revisão bibliográfica a respeito do sistema a ser criado, apontando os componentes do sistema e suas funcionalidades. Ainda neste capítulo é abordado os tipos de comunicações utilizados para interligação dos dispositivos do sistema. O quarto capítulo exibe o esquemático e placas de circuito impresso do sistema que foi utilizado para criação do protótipo. E por fim, o último capítulo abrange as principais conclusões obtidas a respeito do trabalho desenvolvido.

2 AUTOMAÇÃO RESIDENCIAL

Segundo CEDOM (2013): A automação residencial é também chamada domótica, cuja palavra consiste da junção entre casa em latim “domus” com a palavra robótica. É o conjunto de técnicas de controle e monitoramento que proporciona o gerenciamento de todos os recursos habitacionais.

Domótica, notação largamente empregada na Europa, pode também ser explicada segundo publicação feita pela Associação Espanhola de Domótica (Cedom, 2012): é a automatização e o controle aplicados à residência. Esta automatização e controle se realizam mediante o uso de equipamentos que dispõem de capacidade para se comunicar interativamente entre eles e com capacidade de seguir as instruções de um programa previamente estabelecido pelo usuário da residência e com possibilidades de alterações conforme seus interesses. Em consequência, a domótica permite maior qualidade de vida, reduz o trabalho doméstico, aumenta o bem-estar e a segurança, racionaliza o consumo de energia e, além disso, sua evolução permite oferecer continuamente novas aplicações.

Esta tecnologia começou a ser definida nos anos 70 por um grupo de pesquisadores britânicos a partir da criação de um protocolo com o objetivo de comunicar dispositivos entre si. Este protocolo se chama X10 e ainda é utilizado nos dias de hoje.

As indústrias dominam esta área da modernização devido aos benefícios proporcionados. As tarefas passaram a ser realizadas automaticamente através de controladores programados reduzindo acidentes e falhas de equipamentos, trazendo maior segurança aos usuários, gerenciando o consumo de energia, aumentando a produtividade e eficiência. Este resultado positivo altamente difundido para o ramo industrial tornou-se base para implementação de uma casa inteligente, onde as principais vantagens são conforto, qualidade de vida, segurança, economia de energia, praticidade e valorização do imóvel.

A implementação da domótica traz o aumento da qualidade de vida, uma vez que proporciona facilidade na execução de tarefas diárias a medida em que estas podem ser programadas para serem executadas de maneira corriqueira sem mesmo precisar de intervenção do usuário, gerando assim maior sensação de conforto.

Uma das principais necessidades do ser humano é a de se sentir seguro, tanto quanto ao seu patrimônio quanto à sua saúde física. A presença de um sistema que possibilite o controle de variáveis monitoradas como por exemplo presença, iluminação, alarmes de invasão e de incêndio, vazão de gás e água, imagens de segurança, controle de acesso, entre outras, resultam em um domicílio mais seguro. Existe também a necessidade de implementação de um sistema seguro para a prevenção de acidentes com pessoas que precisam de cuidados especiais, como idosos e deficientes físicos.

Outro fator benéfico deste tipo de sistema é a possibilidade do gerenciamento de energia, onde é possível se realizar o controle de dispositivos que consomem maior energia, tais como aquecimento, ar condicionado, iluminação, fazendo com que seu funcionamento seja feito somente quando necessário, evitando assim desperdícios.

Em termos práticos pode-se dizer que a automação residencial proporciona a praticidade de substituir as ações humanas por ações pré-programadas implementadas automaticamente pelo sistema. Dentre elas destacam-se: acionamento automático de luzes através de sensores de presença; controle de intensidade de luminosidade; controle de sistemas de irrigação, motores e cortinas elétricas; climatização; monitoramento de ambientes com o uso de câmeras, acionamento de alarmes de invasão ou de controle ao incêndio; controle de acesso, entre outros itens.

Além de todos os benefícios já citados, no mercado atual onde é crescente a busca por tecnologia e modernização, a indústria da automação residencial traz perspectiva de valorização de imóveis como casas inteligentes.

A Figura 1 resume os elementos que podem ser controlados e monitorados em uma casa.

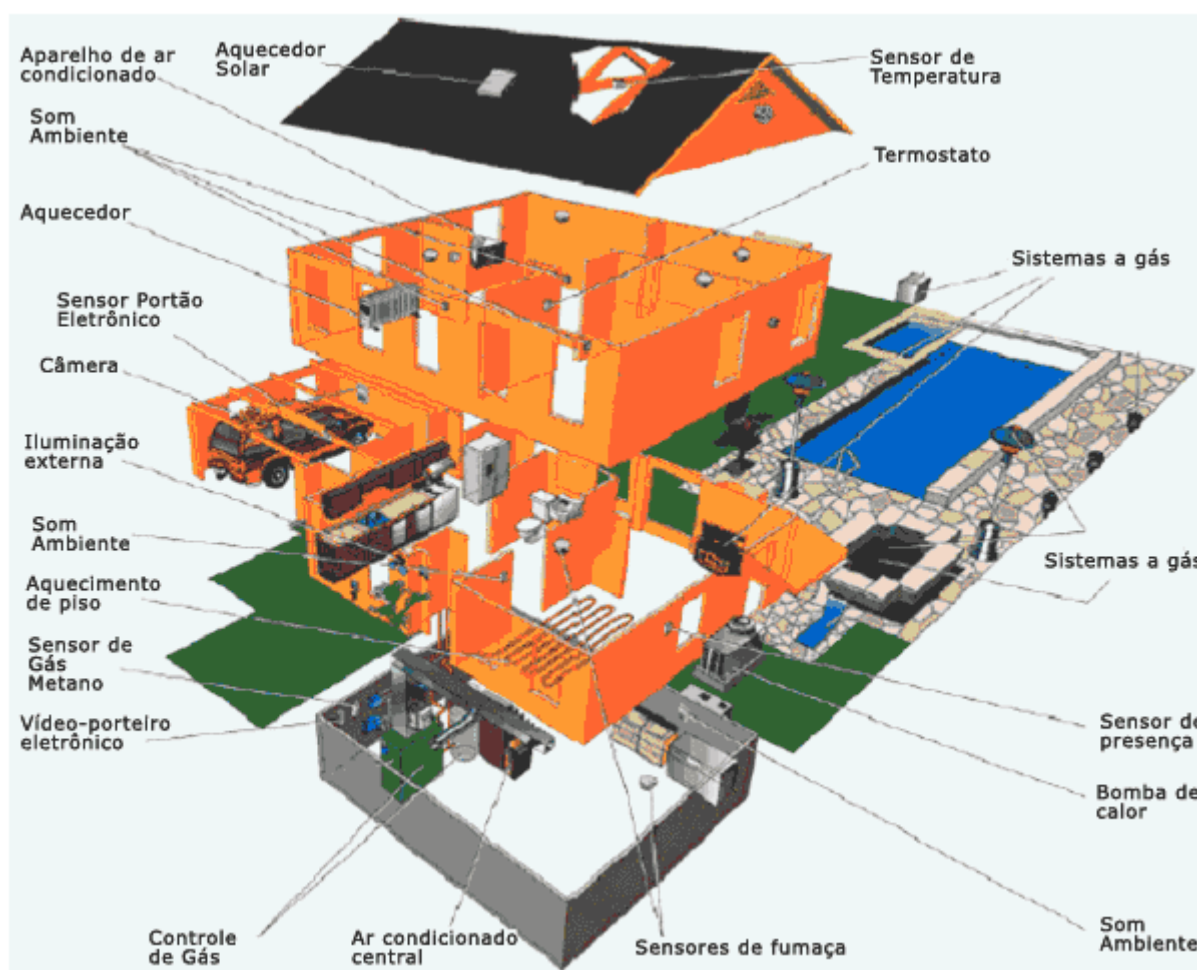


FIGURA 1– ILUSTRAÇÃO DAS VARIÁVEIS A SEREM CONTROLADAS
 FONTE: TECMUNDO (2011).

Segundo informações retiradas do artigo sobre automação residencial feito por Accordi e Dodonov (2012) foram desenvolvidos diferentes tipos de protocolos, ou seja, padrões de comunicações para a automação residencial, alguns destes foram criados especificamente para automação residencial e outros foram adaptados de outros setores, como por exemplo do setor automobilístico. Os protocolos encontrados são: X-10, CEBus, ZigBee, DALI, entre outros.

O X-10 é um protocolo de somente envio, utilizando a rede elétrica como meio de comunicação, a informação enviada de forma binária é transmitida sempre que o sinal senoidal de tensão elétrica passa pelo zero.

O CEBus trata-se de um protocolo que segue o modelo OSI, porém das 7 camadas que este modelo divide, são usadas apenas as camadas física, rede, aplicação e enlace. Os meios físicos que podem ser utilizados com este protocolo são:

rede elétrica, par trançado, cabo coaxial, infravermelhos, rádio frequência e fibra óptica.

O ZigBee é um protocolo que diferente do X-10, envia e recebe os dados sem o uso de fios a frequências de 868 MHz, 915 MHz e também de 2,4 GHz. Os equipamentos são flexíveis, de fácil instalação e econômicos no ponto de vista de consumo de energia.

DALI (*Digital Addressable Lighting Interface*) é um protocolo de comunicação utilizado para o controle unicamente de iluminação. É constituído de um cabo de dois fios com fluxo de informação bidirecional (Teixeira, 2011). A norma IEC 60929 especifica o padrão internacional onde independente dos fabricantes dos produtos, o protocolo assegura a intercambialidade e a interoperabilidade dos dispositivos. O protocolo DALI não pode ser usado para controle de outros sistemas que não sejam de iluminação porém, é eficaz para a seleção de informações como por exemplo a ocorrência de um defeito em uma luminária, fazendo-se útil o seu interfaceamento com sistemas automáticos em edifícios quando são necessários relatórios de supervisão remota e emissão de ordens de serviço. Cada unidade que se deseja controlar com o uso deste protocolo recebe um endereço, assim é possível se comunicar com cada componente.

3 IMPLEMENTAÇÃO DO PROJETO

Um sistema de automação residencial é composto por uma rede de comunicação que integra diferentes dispositivos eletrônicos que têm por função o monitoramento e controle dos ambientes, propiciando todas as aplicações descritas no item anterior. Os elementos constituintes deste sistema serão: Equipamentos, comunicação, supervisor e programação. O diagrama esquemático do projeto pode ser visto na Figura 2.

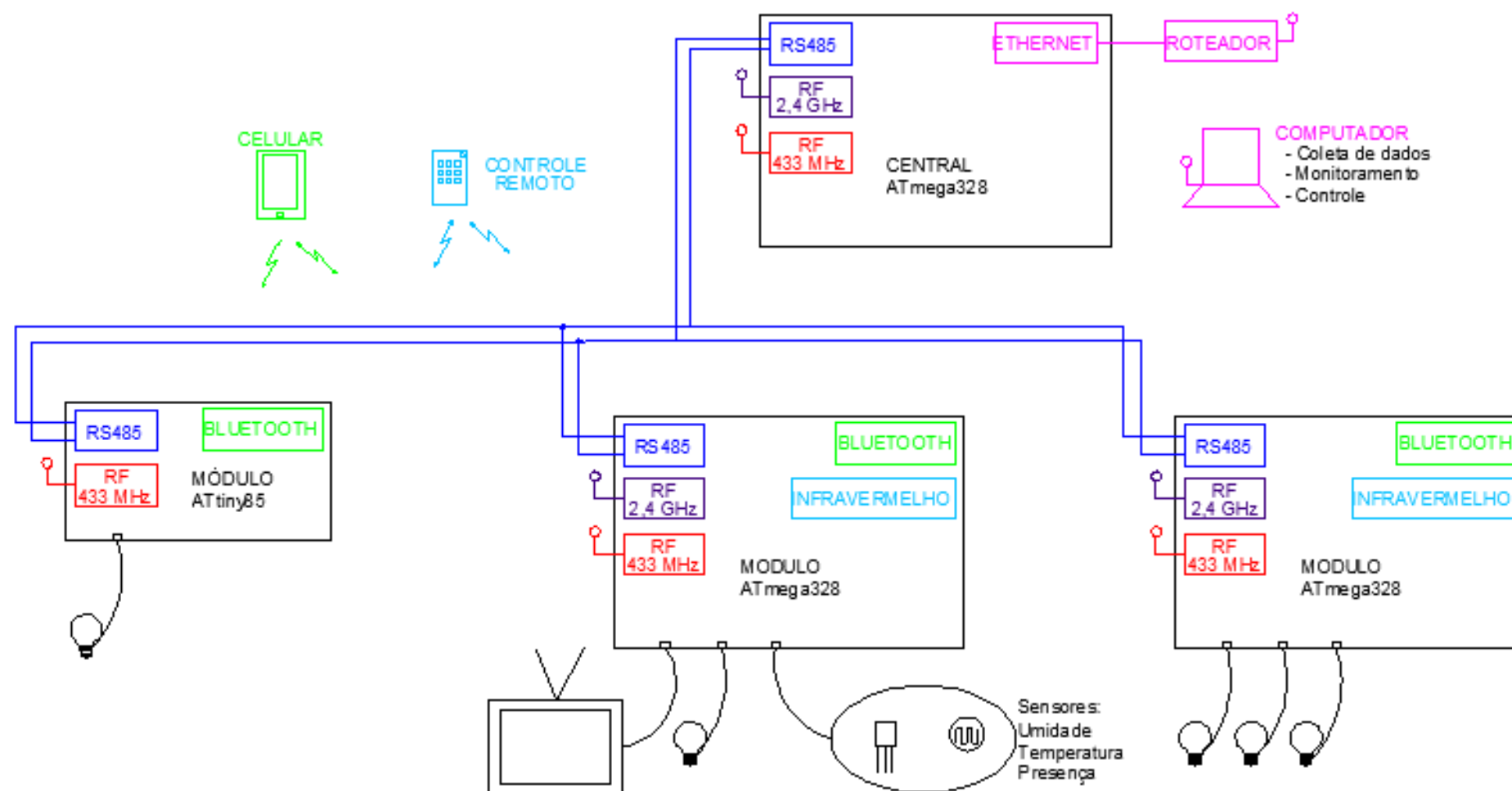


FIGURA 2 - ESQUEMÁTICO DO SISTEMA
FONTE: O AUTOR (2014).

O sistema que este trabalho propõe terá por finalidade a integração de todos os dispositivos necessários para se realizar o gerenciamento e controle de um domicílio.

Cada ambiente de uma casa possuirá um módulo controlado por ATmega328 ou Attiny85 a fim de fazer o controle de lâmpadas, televisões, cortinas, entre outros. Além dos módulos acoplados em cada ambiente, também é necessário acoplar uma central que será capaz de armazenar todos os dados do sistema.

Para realizar a integração entre todos os dispositivos optou-se por estudar diferentes tipos de comunicação tais como: padrão serial RS485, ondas de rádio frequência, *bluetooth*, protocolo *ethernet* e raios infra-vermelho. Com esta diversidade de comunicações o sistema é preparado para as diferentes situações nos domicílios. Por exemplo o controle de portão, luminárias, aquecimento de piscina, ou seja, equipamentos que estão em um ambiente externo, sem conexão com o interior da residência.

Cada módulo terá a possibilidade de se comunicar com outro módulo e também com a central.

A central estará conectada à internet através da comunicação ethernet com a finalidade de, através de um supervisor, ter a possibilidade de controlar e monitorar toda a residência, podendo o usuário interferir nos equipamentos de sua residência mesmo não estando próximo a ela. Para realização desta operação, a central também se comunicará com os módulos de forma ativa, ou seja, enviando comandos.

Além do uso do supervisor o usuário poderá fazer o controle de sua residência através de um controle remoto que emita raios infra-vermelho e também através de um celular, *tablet* com o uso do *bluetooth*.

3.1 EQUIPAMENTOS

Para o desenvolvimento do controle e monitoramento de uma residência, é necessário escolher adequadamente os equipamentos que compõe o sistema. O microcontrolador terá o papel de comandar os sensores e atuadores por meio de lógicas pré-definidas. O usuário dará as ordens ao microcontrolador que por sua vez

fará o processamento das medições feitas pelos sensores e também realizará ações por meio dos atuadores.

3.1.1 Atuadores

Os atuadores serão os elementos físicos mais próximos do equipamento a ser operado, necessários para que as funções ordenadas pelo morador sejam traduzidas em ações. Como exemplo de atuadores temos: relé, solenoides, contatores, entre outros.

Estes dispositivos são basicamente interruptores eletromecânicos. O microcontrolador enviará um sinal para mudança de estado destes interruptores, acionando ou desligando os equipamentos ligados a ele.

Os itens que podem ser controlados são cortinas, portões elétricos, eletrodomésticos, ar-condicionado, avisos sonoros de alarme, avisos visuais, lâmpadas.

3.1.1.1 Relé

O exemplo mais comum a ser citado é o acionamento de uma lâmpada com uso de relé, conforme Figura 3.

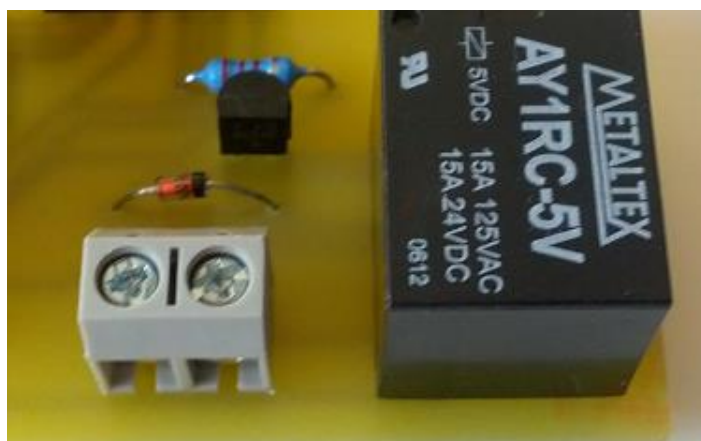


FIGURA 3 - ESQUEMÁTICO USO DE RELÉ
FONTE: O AUTOR (2014).

Ao conectar um relé a uma lâmpada, este funcionará como um interruptor, quando o residente desejar acender a luz, o microcontrolador receberá o comando e mandará, através de sinal elétrico, o relé fechar os contatos acendendo assim a lâmpada. Como o relé opera com um sinal de tensão de 5Vcc, o microcontrolador é capaz de fornecer a tensão adequada para mudança de estado dos contatos do relé.

De acordo com a ABNT, o relé de proteção é um dispositivo por meio do qual um equipamento elétrico é operado quando se produzem variações nas condições deste equipamento ou do circuito em que ele está ligado, ou em outro equipamento ou circuito associado.

Para o projeto será usado um relé com bobina e contato interno. Ao passar corrente pela bobina, esta induz um campo magnético atraindo um pino interno fechando o contato conforme ilustrado na Figura 4.

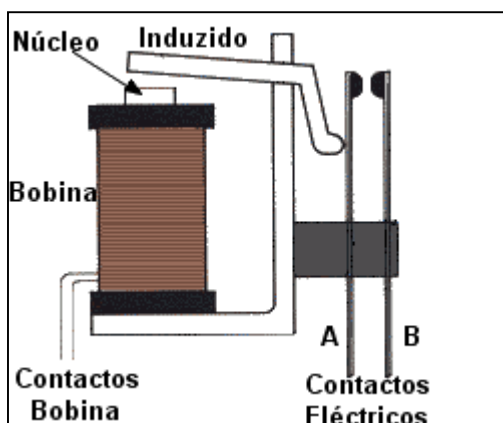


FIGURA 4 - RELÉ
FONTE: ELECTRONICA-PT (2014).

Para fornecer uma corrente adequada ao relé é preciso conectar um transistor a fim de aumentar a corrente fornecida pelo microcontrolador. Também é necessário utilizar um diodo para não ocorrer inversão do sentido de corrente podendo danificar o microcontrolador. Pode-se ver na figura 5, o esquemático composto por transistor, diodo, resistência e o relé.

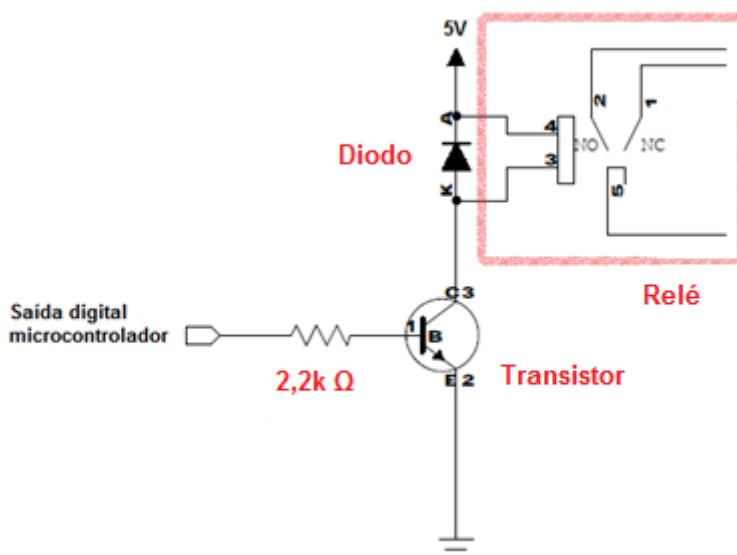


FIGURA 5 - ESQUEMÁTICO MÓDULO RELÉ
FONTE: O AUTOR (2014).

3.1.2 Sensores

Segundo o livro Sensores Industriais (THOMAZINI e ALBUQUERQUE, 2005), sensor é “termo empregado para designar dispositivos sensíveis a alguma forma de energia do ambiente que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza que precisa ser medida, como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc.”

Os sensores são responsáveis pelo monitoramento de grandezas físicas em uma automação residencial, as quais podem ser temperatura, umidade, presença, entre outras, podendo estas serem detectadas pontualmente e transformadas em sinais elétricos que serão tratados pelo controlador.

Existem diversos tipos de sensores e eles podem ser divididos em sensores analógicos e sensores digitais. Os sensores analógicos são baseados em sinais analógicos, ou seja, sinais que variam entre dois valores de tensões. Esta variação pode ou não ser linear dependendo do sensor utilizado. Os sensores digitais utilizam valores de tensões definidos, por exemplo o nível de tensão alto (1) ou nível de tensão baixo (0), não existindo valor intermediário entre estes.

3.1.2.1 Sensor de presença

O sensor de movimento é muito utilizado na domótica, pois este é capaz de detectar movimentos ocorridos em uma determinada área de alcance. Atualmente, em vários edifícios este tipo de dispositivo é utilizado para acendimento automático de luminárias em áreas comuns, outro uso bastante empregado é no quesito de segurança, onde o sensor aciona um alarme caso ocorra o movimento de um objeto ou de uma pessoa.

O método de medição de intrusão é feito por sensores que utilizam infravermelho, como o PIR, que é um dispositivo eletrônico capaz de detectar a presença de pessoas ou animais através da diferença do calor emitido pelo corpo em questão e o espaço ao redor. Para obter uma melhor cobertura da região em que está inserido o PIR tem uma lente de Fresnel que distribui os raios infravermelhos em diferentes zonas horizontais ou inclinadas aumentando assim sua área de atuação. Na Figura 6 encontra-se o sensor PIR.



FIGURA 6 - SENSOR DE PRESENÇA PIR
FONTE: FUTURLEC (2014).

Outro dispositivo utilizado para detecção de presença é feito com o uso do sensor PIR associado ao sensor ultrassônico, com o uso de dois sensores diferentes é feita dupla detecção para evitar alarmes falsos. O funcionamento do sensor ultrassônico ocorre com a ativação do cristal de quartzo, que emite ondas ultrassônicas em frequências entre 25 e 40kHz para realizar a detecção de presença. Essas ondas emitidas são comparadas com a frequência do sinal refletido (efeito Doppler) e a diferença entre elas é interpretada como presença no espaço de cobertura do sensor (Silva, 2009).

3.1.2.2 Sensor de temperatura

Existem diversos tipos de sensores de temperatura empregados em automação residencial, cada um com sua característica específica para se empregado em diferentes casos, como por exemplo: medições de temperatura em ambientes abertos onde existe a necessidade de um encapsulamento no sensor, medições de temperatura com maior exatidão, medições de temperatura em faixas diferentes das temperaturas ambientes, entre outras.

Um sensor utilizado em projetos eletrônicos é o sensor de temperatura LM35. Este sensor é empregado na medição de temperaturas ambientes devido a sua faixa de medição, entre -55°C a 150°C , e também por sua precisão.

O sensor faz uma medição da temperatura ambiente e fornece uma tensão analógica conforme a temperatura medida. Ao conectar um microcontrolador a este sensor, o microcontrolador poderá fazer a análise dessa tensão analógica fornecida pelo sensor, transformando assim em um valor em graus Celsius compreensível pelos moradores.

Outro sensor que também faz medição de temperatura é o sensor DHT22 que tem a capacidade de realizar a medição de temperatura e também de umidade.

3.1.2.3 Sensor de umidade

Um exemplo de sensor de umidade é o sensor DHT22, também chamado de AM2302. Sua saída de sinal calibrada é digital, dependendo de um microcontrolador para a análise do sinal.

Pode-se ver na Figura 7 o sensor de umidade.

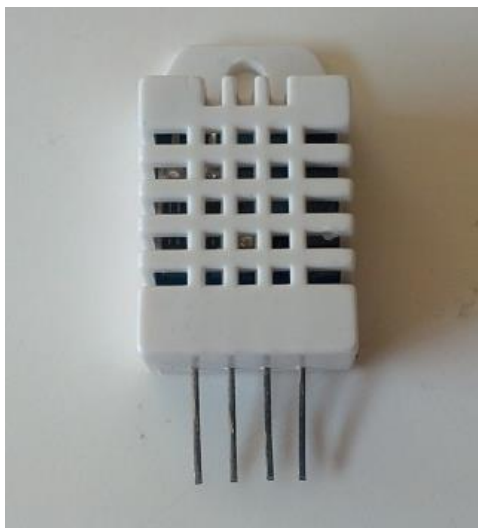


FIGURA 7 - SENSOR DE UMIDADE
FONTE: O AUTOR (2014).

A alimentação deve ser entre 3.3 e 6 Vcc e a corrente, durante a transmissão do sensor com o microcontrolador, é de 2,5mA. Suas especificações podem ser encontradas no *datasheet* no anexo A.

O uso de um sensor de umidade em conjunto com o sensor de temperatura pode futuramente auxiliar na criação de um sistema automático de climatização de ar.

Para este projeto foi usado a biblioteca chamada DHT.h criada pela empresa *Adafruit* a fim de se obter os valores de temperatura e umidade do sensor. Ao realizar a programação é necessário digitar para se obter os valores de umidade e temperatura os seguintes comandos criados pela empresa (ADAFRUIT, 2011):

```
float u = dht.readHumidity();  
float t = dht.readTemperature();
```

No caso apresentado as variáveis *u* e *t* receberam a atribuição dos valores de umidade e temperatura. A partir deste momento pode-se usar essas variáveis para enviar para a central os valores medidos.

3.2 MICROCONTROLADOR

Na domótica, o microcontrolador é um dos principais dispositivos, é ele que irá formar o sistema adquirindo dados dos sensores e controlando os atuadores através de um supervisor que será explicado no capítulo 3.6.6.

O microcontrolador ou MCU (*micro controller unit*) é um dispositivo que possui todas as funcionalidades de um computador, porém estão encapsulados em um único chip. Os componentes essenciais formadores são:

- Unidade central de processamento (CPU);
- Memórias de dados e de programa;
- Barramentos de dados, instruções e de controle;
- Portas de entrada e de saída (E/S);
- Periféricos tais como conversores analógicos digitais, osciladores internos.

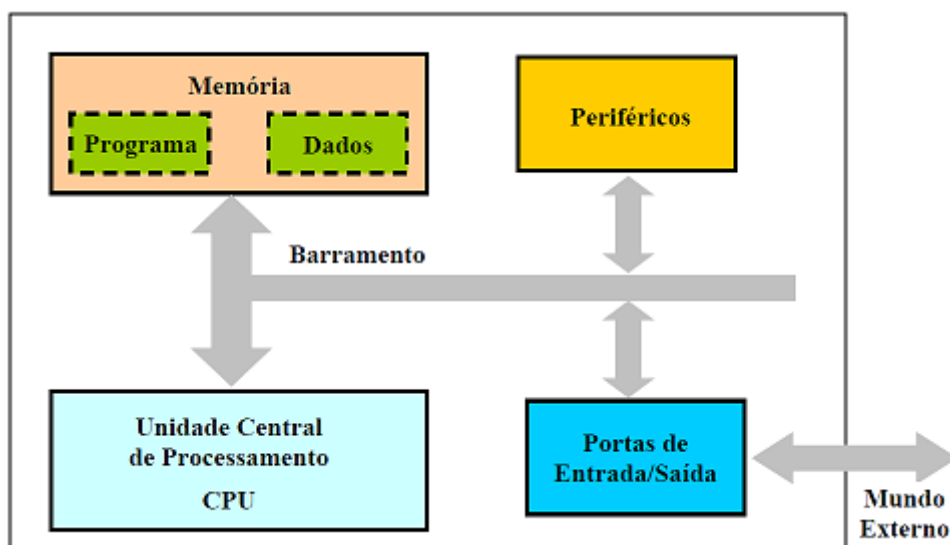


FIGURA 8 - ESTRUTURA DE UM MICROCONTROLADOR
FONTE: PASTRO (2012).

Os microcontroladores utilizados exigem uma frequência de *clock* inferior aos microprocessadores modernos, mas sua utilização é satisfatória para este projeto. O seu consumo de energia é pequeno assim como seu custo por possuir memória e capacidade de processamento inferiores as de um computador do tipo *desktop*. Os microcontroladores não possuem conectores de entrada e saída, acessórios como botões, *leds*, *displays* e por isso é necessário montar um kit de desenvolvimento. O kit de desenvolvimento que vem sendo cada vez mais utilizado é o Arduino, mostrado na Figura 9. (ARDUINO, 2013)

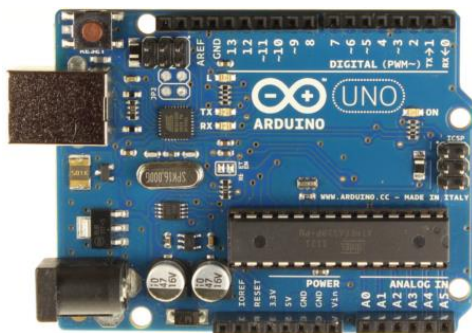


FIGURA 9 - KIT DE DESENVOLVIMENTO ARDUINO UNO
FONTE: ARDUINO (2013).

Os diversificados tipos de MCU se diferem em suas propriedades físicas, tais como: quantidade de memória, tensão de alimentação, velocidade de processamento, arquitetura física e instruções. Para o projeto será utilizado o *chip* do próximo item.

3.2.1 ATmega328

Será utilizado para o protótipo o microcontrolador ATmega328, conforme Figura 10.



FIGURA 10 - ATMEGA328-PU
FONTE: O AUTOR (2014).

O ATmega328 é um microcontrolador 8 bit da família AVR e sua pinagem pode ser vista na figura 11.

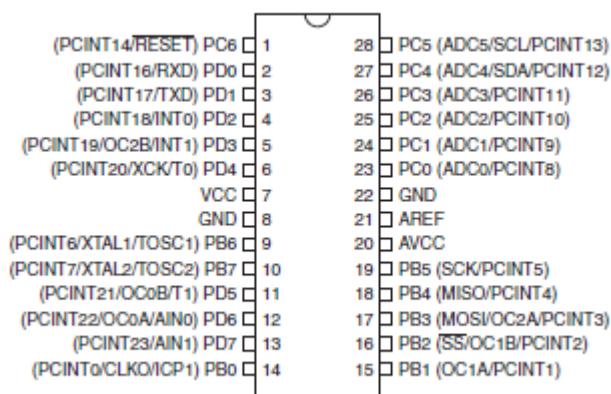


FIGURA 11 - PINAGEM ATMEGA328-PU
FONTE: ATMEL (2014).

O ATmega328 é um microcontrolador de alta performance, porém com baixo consumo de energia. A execução das funções são feitas em um único ciclo de *clock*. Dos 28 pinos, possui 14 pinos digitais de entrada e saída, onde 6 destes pinos podem ser usados como saída PWM. Existem também 6 pinos de entradas analógicas. A tensão de operação é de 5 Volts. Sua memória *flash* é de 32KB onde 0.5KB são usados pelo *bootloader*. Também possui 2KB de SRAM, 1KB de EEPROM e um *clock* de 16MHZ ao acoplar um cristal, (ATMEL, 2014).

Para o protótipo do sistema de domótica o uso deste microcontrolador possibilitará a leitura da medição de sensores e o acoplamento de atuadores.

3.2.2 ATtiny85

O também microcontrolador ATtiny85 (Figura12) será empregado no sistema a fim de se observar como se comporta este tipo de chip visto que sua capacidade de pinos é reduzida se comparado ao ATmega328.

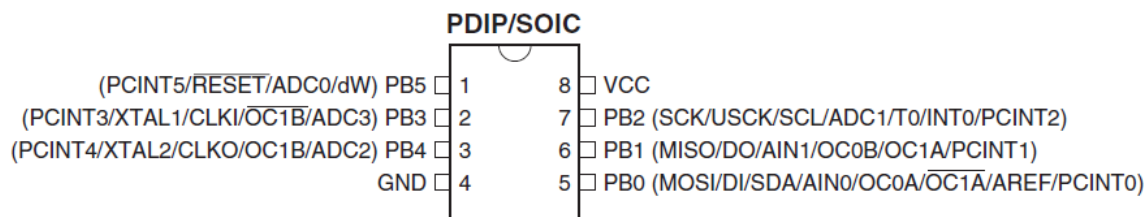


FIGURA 12 - PINAGEM ATTINY85
FONTE: ATMEL (2014).

Este *chip* possui um encapsulamento com 8 pinos e um oscilador interno de 20Mhz. Sua memória *flash* é de 8 Kbytes para armazenar o código, 512 bytes de SRAM interna para processar as informações e 521 bytes de EEPROM para armazenar os dados. Dos oito pinos, seis são pinos de entrada e saída, sendo o restante pino de alimentação 2,7 a 5,5 Vcc e terra, (ATMEL,2014).

3.3 FONTE DE ALIMENTAÇÃO

Todos os componentes do sistema precisam ser alimentados com um valor de tensão adequado. Os dispositivos que serão acoplados nos ATmega328 do sistema operam com uma tensão entre 3,3 Vcc e 5 Vcc. Para o microcontrolador fornecer estas tensões é necessário o uso de uma fonte de alimentação externa entre os limites de 7 Vcc a 12 Vcc.

3.4 CENTRAL

A central consiste de um microcontrolador ATmega328 capaz de se conectar à internet a fim de se fazer o controle e monitoramento via web e também é ela que irá receber informações vindas dos módulos tais como informações dos estados dos equipamentos e dos sensores. Ao receber um comando via supervisor do usuário, a central deverá enviar dados para os módulos através de comunicação serial RS485 e rádio frequência de 433Mhz e 2,4Ghz.

A central é composta pelos seguintes componentes:

- Kit de desenvolvimento Arduino UNO
- Módulo Ethernet W5100
- Fonte de alimentação
- MAX485 (RS485)
- Transmissor e receptor RF 433Mhz
- nRF24L01 2,4Ghz

No quarto capítulo será abordado a montagem e esquemático deste equipamento.

3.5 MÓDULO

Para montagem dos módulos ocorrerá duas situações distintas. A primeira situação é com o uso do ATmega328 e a segunda com o uso do ATtiny85.

Os módulos que serão microcontrolados por um ATmega328 e serão responsáveis por cada ambiente em que estiverem instalados poderão ser controlados via Bluetooth e raios infravermelho. Este módulo é composto pelos seguintes componentes:

- ATmega328
- Fonte de alimentação
- MAX485 (RS485)
- Transmissor e receptor RF 433Mhz
- nRF24L01 2,4Ghz
- Bluetooth
- Infravermelho
- Sensores
- Atuadores

Com o uso do ATtiny85 o usuário terá um menor número de facilidades devido a capacidade deste microcontrolador, por este motivo o morador terá somente a possibilidade de controlar seu cômodo através de Bluetooth, ou pelo supervisor da internet. A composição deste módulo será:

- ATtiny85
- Fonte de alimentação
- Bluetooth ou rádio frequência ou RS485
- Atuadores ou sensores

No quarto capítulo será abordado a montagem e esquemático deste equipamento.

3.6 COMUNICAÇÃO

O sistema de domótica a ser implementado será constituído de uma central microcontrolada que se comunica enviando comandos e recebendo dados com vários módulos também microcontrolados, distribuídos por diferentes ambientes de uma mesma habitação.

Serão estudados diferentes tipos de comunicação entre central e módulo: por fio, por rádio frequência de 433Mhz e 2,4Ghz.

Os tipos de comunicação utilizados pelos usuários para acionar os equipamentos serão: *bluetooth*, *ethernet* e infravermelho.

3.6.1 RS485

O método mais usualmente utilizado para comunicação entre dispositivos é feito com o uso de cabos. Os cabos fazem a conexão através de um barramento, assim todos os dispositivos que forem capazes de realizar esta comunicação podem ser conectados ao barramento.

O microcontrolador não é capaz de fornecer uma corrente adequada para envio de dados a uma longa distância, para isso é necessário utilizar um dispositivo para aumentar a potência do sinal a ser enviado.

O dispositivo escolhido para realizar esta comunicação é o MAX485, conforme figura 13.

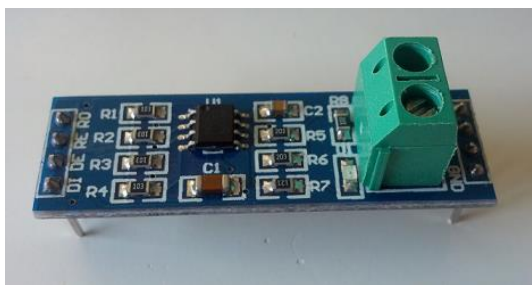


FIGURA 13 - MAX485
FONTE: O AUTOR (2014).

O módulo MAX485 utiliza o padrão de comunicação RS-485, também conhecido como EIA-485, o RS é advindo da abreviação de *Recommended Standard*.

Este nome é uma padronização criada pelo comitê *Electronic Industries Association* onde o objetivo era de padronizar uma interface comum de comunicação de dados entre vários componentes. Mais informações sobre o MAX485 podem ser vistas no anexo B.

A comunicação RS485 é feita pela diferença entre as tensões que as linhas transmitem, resultando em 1 ou 0. Pode ser feita a comunicação *half-duplex* e *full-duplex*. A comunicação *half-duplex* é um método de envio de dados onde eles trafegam somente em um sentido por vez, assim, a capacidade total da linha pode ser usada. Por outro lado, a comunicação *full-duplex* é um método de envio de dados simultâneo, ou seja, os dados podem ser enviados e recebidos ao mesmo tempo. Para o projeto será utilizado a comunicação *half-duplex*.

Existem 4 terminais (DE, RE, GND e VCC) no módulo MAX485 e um borne de saída (A, B). Os terminais VCC e GND são a alimentação de 5Vcc e o terra respectivamente, para transmissão é necessário alimentar com 5Vcc os terminais DE e RE e com o GND nos terminais RE e DE do receptor.

Para realização da comunicação entre os diversos módulos e a central foi usada a biblioteca VirtualWire.h. Esta biblioteca tem por finalidade o envio de pequenas mensagens, sem endereçamento utilizando modulação ASK, ou seja, o sinal digital a ser transmitido é variado em função da alteração da amplitude da onda portadora. Esta onda portadora é comutada entre dois valores, usualmente ligado e desligado, 1 ou 0 . O uso da modulação ASK serve como sincronização entre o transmissor e o receptor.

O comando utilizado para envio dos dados após definida a mensagem a ser entregue é:

```
vw_send((uint8_t *)msg, strlen(msg));
```

A leitura da mensagem que foi entregue é feita através de armazenamento dos dados (&buflen) em um vetor de tamanho variável conforme a mensagem. Para programação é necessário que o vetor de armazenamento dos dados esteja em uma função condicionante SE, esta função irá analisar se existe alguma mensagem recebida. Caso exista esta mensagem o microcontrolador será capaz de informar cada dado desta mensagem que foi armazenada em um vetor:

```
if (vw_get_message(buf, &buflen))
{
    mensagem = &buflen;
```

}

3.6.2 RÁDIO FREQUENCIA

A comunicação feita por rádio frequência não utiliza cabos entre os equipamentos. São utilizados rádios para fazer a transmissão e o recebimento dos dados por meio de campos ou ondas eletromagnéticas ponto a ponto.

Rádio frequência são sinais que se propagam por um condutor cabeado, normalmente cobre, e são irradiados no ar através de uma antena. Uma antena converte um sinal do meio cabeado em um sinal *wireless* (sem fio) e vice-versa. Os sinais irradiados no ar livre, em forma de ondas eletromagnéticas, propagam-se em linha reta e em todas as direções (SENAI,2014).

Para o projeto serão utilizadas duas faixas de frequência distintas: 433Mhz e 2,4Ghz.

3.6.2.1 Rádio frequência 433Mhz

Para a comunicação entre a central e os módulos, o uso de rádio frequência será um dos métodos utilizados. Com este método os módulos poderão ser instalados com mais facilidade em locais mais afastados e em locais onde os eletrodutos não se interligam com o cômodo onde a central estará localizada, por exemplo em portões que estão localizados fora da casa onde o acesso por fio é difícil e dispendioso de ser realizado.

O módulo a ser adotado para o uso da faixa de frequência de 433MHz pode ser visto na Figura 14:

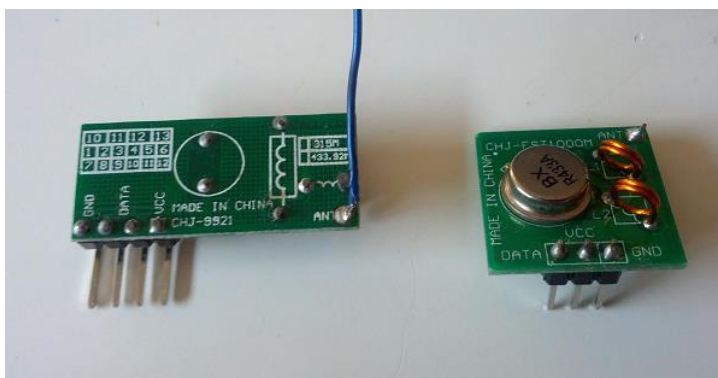


FIGURA 14 - RECEPTOR E TRANSMISSOR RF433MHZ
FONTE: O AUTOR (2014).

A modulação em amplitude (AM), que na aceção de Haykin e Moher (2011), é a forma com que o módulo realiza a transmissão de dados, ou seja, a portadora tem sua amplitude variada conforme a informação que se deseja transmitir.

Segundo dados retirados do fabricante (ON SHINE ENTERPRISE, 2010) o alcance de transmissão é de até 200 metros. A taxa de transferência é de 4KB/s com uma potência de transmissão de 10mW. O módulo de recepção opera com uma sensibilidade de -105dB.

Na figura 14 pode ser visto a direita o módulo de transmissão, e na esquerda o módulo de recepção. Para utilização destes módulos basta alimentá-los com 3,3Vcc, acoplá-los ao terra e por fim enviar o sinal digital ao módulo. Este sinal digital será amplificado pelo módulo e enviado via ondas eletromagnéticas ao módulo receptor.

O envio e recebimento de dados com o uso destes módulos também é feito via biblioteca VirtualWire.h.

Para se obter o alcance máximo é necessário a implementação de antenas. O comprimento da antena é calculado em função da saída do amplificador e o meio físico que será transmitido o sinal, no caso o ar. Utilizando uma haste vertical, o comprimento da antena é 1/4 do comprimento de onda da frequência envolvida. O valor de 1/4 é uma constante que representa uma onda senoidal quando passa de um meio para outro.

O comprimento de onda é:

$$\lambda = \frac{c}{f} \quad (3.1)$$

onde

λ : é lâmbida, ou seja, comprimento de onda, em metros;

c: é a velocidade da onda eletromagnética no meio 300×10^8 elevado a 8 m/s;
f: é a frequência.

Para frequência igual a 433Mhz obtemos:

$$\lambda = \frac{300 \times 10^8}{433 \times 10^8} \quad (3.2)$$

$$\lambda = 0,69284m \quad (3.3)$$

Com isso, o comprimento da antena é igual ao comprimento de onda dividido por 4:

$$Antena = \frac{\lambda}{4} \quad (3.4)$$

$$Antena = \frac{0,69284}{4} \quad (3.5)$$

$$Antena = 17,3 \text{ centímetros} \quad (3.6)$$

3.6.2.2 Rádio frequência 2,4Ghz

Outra possibilidade de implementação de comunicação por rádio frequência é a utilização de módulos NRF24L01, esse único módulo possui a capacidade de transmissão e recepção de ondas eletromagnéticas na faixa de frequência de 2,4GHz. As características do módulo NRF24L01 podem ser vistas no Anexo C.



FIGURA 15 - NRF24L01
FONTE: O AUTOR (2014).

Este dispositivo de transmissão e recepção de sinais é utilizado em aplicações industriais permitindo comunicação de vários pontos de modo a formar uma rede completa.

A capacidade de alcance de recepção e envio de sinal é de 100 metros, com um consumo pequeno de energia. Não há necessidade de acoplar uma antena. Este módulo possui protocolo de comunicação *ShockBurst* para transmissão de dados.

3.6.3 Bluetooth

Para o usuário comandar a residência inteligente, um dos métodos utilizados para comunicação entre os dispositivos móveis e o módulo é o uso de *bluetooth*. Para isso são utilizados receptores que usam a tecnologia *bluetooth*. No sistema escolhido os dispositivos móveis (celulares ou *tablets*) serão os responsáveis pela transmissão de dados.

A tecnologia Bluetooth é um dispositivo de curto alcance, cujo objetivo é eliminar os cabos nas conexões entre dispositivos eletrônicos, tanto portáteis como fixo. As principais características desta tecnologia são suas confiabilidades, baixo consumo e mínimo custo. Várias das funções das especificações é opcional, o que permite a diversificação dos produtos. Dispositivos Bluetooth operam na faixa ISM (Industrial, Scientific, Medical) centrada em 2,45 GHz em sua camada física de Rádio (RF) que era formalmente reservada para alguns grupos de usuários profissionais. Nos Estados Unidos, a faixa ISM varia de 2400 a 2483,5 MHz. Na maioria da Europa a mesma banda também está disponível. No Japão a faixa varia de 2400 a 2500 MHz. Os dispositivos são classificados de acordo com a potência e alcance, em três níveis: classe 1 (100 mW, com alcance de até 100 m), classe 2 (2,5 mW e alcance até 10 m) e classe 3, (1 mW e alcance de 1 m, uma variante muito rara). Cada dispositivo é dotado de um número único de 48 bits que serve de identificação. Os dispositivos Bluetooth se comunicam entre si e formam uma rede denominada piconet, na qual podem existir até oito dispositivos interligados, sendo um deles o mestre (master) e os outros dispositivos escravos (slave); uma rede formada por diversos "masters" (com um número máximo de 10) pode ser obtida para maximizar o número de conexões. A banda é dividida em 79 portadoras espaçadas de 1 Megahertz, portanto cada dispositivo pode transmitir em 79 diferentes frequências; para

minimizar as interferências, o dispositivo "master", depois de sincronizado, pode mudar as frequências de transmissão dos seus "slaves" por até 1600 vezes por segundo. Em relação à sua velocidade pode chegar a 3 Mbps em modo de transferência de dados melhorada (EDR) e possui três canais de voz. Toda transferência de dados se dá no canal físico que se subdivide em unidades de tempo, denominadas ranhuras. E os dados intercambiados entre os dispositivos transitam em forma de pacotes, estes por sua vez deverão chegar a estas ranhuras para que a transmissão de dados ocorra com sucesso. Uma das características da tecnologia Bluetooth é a capacidade de transmissão de dados bidirecional, e isso se deve a técnica por ela utilizada de múltiplo acesso ou Duplex por divisão de tempo (TDD). Sobre o canal físico podemos dizer que é composto por uma camada de enlace físico e canais com seus devidos protocolos de controle. A hierarquia abaixo e acima dos níveis de enlaces é a seguinte: Canal físico, enlace físico, comunicação lógica, enlace lógico e canal L2CAP. (BONATTO e OLIVEIRA, 2013)

Neste sistema será criado uma aplicação para dispositivos móveis com sistema operacional Android. Este aplicativo será o responsável por utilizar a tecnologia de transmissão *bluetooth* para enviar dados, por exemplo do celular para o módulo acoplado ao microcontrolador. O microcontrolador irá fazer o tratamento dos comandos recebidos pelo celular. O dispositivo *bluetooth* receptor recebe os dados e envia via serial ao microcontrolador. O Apêndice H mostra como foi feito o aplicativo.

3.6.4 Ethernet

O acesso ao supervisor, que será explicado no próximo capítulo, se dará por uma página na web conectada à internet. Esta página se conectará ao microcontrolador central para coletar comandos feitos pelo usuário e mostrar como o sistema está se comportando, ou seja, quais equipamentos estão operando, como estão comportando os sensores.

Para isso, o Arduino Uno deve ter a capacidade de se comunicar via protocolo ethernet com o roteador. Inserindo um módulo ethernet no atmega328, este se conectará a internet.

O módulo escolhido possui um chip Wiznet W5100 com uma biblioteca de network (IP) com suporte aos protocolos TCP e UDP, (WIZNET, 2011).

A conexão entre o ATmega328 e o módulo ethernet é feita por ICSP, método de gravação de dispositivos programáveis. Este método faz com que após a montagem do circuito seja possível fazer a gravação de um programa no chip através da porta serial.

O módulo ethernet se conecta a central e para ter conexão com a internet é necessário utilizar um cabo de rede com conector RJ45, fazendo assim a ligação do Wiznet W5100 com o roteador da residência. Na figura 16 pode-se visualizar o módulo ethernet a ser utilizado no sistema.



FIGURA 16 - MÓDULO ETHERNET
FONTE: O AUTOR (2014).

Com o uso deste módulo deverá ser programado na central as configurações de rede utilizando a biblioteca Ethernet.h.

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192,168,0,99};
byte gateway[] = { 192,168,0,1};
byte subnet[] = { 255,255,255,0};
Server server(8080);
```

A variável declarada como *byte mac* é o endereçamento físico do módulo ethernet a ser utilizado. *Byte IP* representa o endereço de IP que o microcontrolador assumirá e *byte gateway* o endereço de *IP* do roteador. *Byte subnet* representa a máscara de rede. Por fim, a porta do servidor a ser utilizado é escolhida e setada, no caso acima a escolhida foi 8080.

3.6.5 INFRAVERMELHO

O uso de sensores infravermelho é muito empregado nas residências. O controle remoto, por exemplo, funciona através do envio de sinais infravermelho para televisões, ar-condicionado, equipamentos de som a fim de regular a intensidade de volume, regular o brilho, ligar e desligar, acionar portões de garagem.

Para o sistema, o meio de comunicação por infravermelho possibilitará o uso de um controle remoto para acionamento dos equipamentos de cada ambiente.

Deverá ser acoplado aos módulos de cada ambiente um sensor de infravermelho, usualmente chamados de sensores IR e o modelo a ser utilizado no projeto será o TSOP4838.



FIGURA 17 - SENSOR IR
FONTE: O AUTOR (2014).

Primeiro é necessário escolher um controle remoto para envio dos sinais, a partir deste deverá ser analisado e colocado em uma tabela os códigos que são enviados ao apertar de um botão. Com esta tabela de códigos é possível programar qual ação será feita ao ser pressionado os diferentes botões do controle remoto.

Para o uso deste sensor utilizou-se a biblioteca IRremote.h (SHIRRIFF, 2009).

Utilizando a biblioteca ao executar o comando a seguir foi possível obter os valores de forma decimal recebidos pelo sensor.

```
if (irrecv.decode(&results))  
{  
  Serial.println(results.value, DEC);  
}
```

Nos módulos de cada ambiente será programado no ATmega328 o que cada código executará de função. Pode-se ver a seguir o que cada código de cada botão apertado do controle executa no ambiente.

TABELA 1 – CÓDIGOS ENVIADOS PELO CONTROLE

Botão do controle	Código recebido
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
0	0

FONTE: O AUTOR, 2014.

Os códigos que o controle remoto enviou são exatamente o número do teclado alfanumérico do controle. Com esta tabela pode-se definir o que cada botão irá executar no ambiente em que estiver instalado o sensor infravermelho.

TABELA 2 – CÓDIGO DO CONTROLE E SUA FUNÇÃO NO SISTEMA

Botão do controle	Código Recebido	Ligar/ Desligar Equip. 1	Ligar/ Desligar Equip. 2	Ligar/ Desligar Equip. 3
1	1	X		
2	2		X	
3	3			X

FONTE: O AUTOR, 2014.

3.6.6 SUPERVISÓRIO

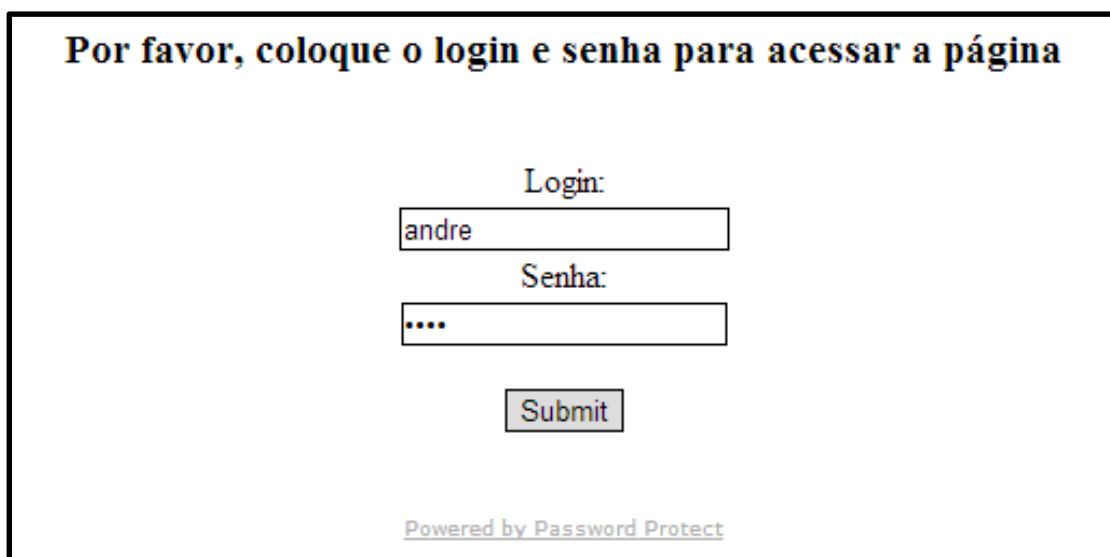
Todo o sistema será monitorado e controlado por um supervisor, ou seja, será implementada uma interface gráfica no qual o usuário poderá visualizar como está se comportando sua residência assim como controlá-la.

Para realização do supervisor, optou-se por desenvolver uma página na *web* (*World Wide Web*), que está conectada à internet e pode ser acessada por computadores, ou por quaisquer outros dispositivos com acesso à internet, em qualquer lugar do mundo.

A base de linguagem de programação PHP será usada para comunicação entre a página da web e o microcontrolador. Esta linguagem possui código fonte livre. O código para criação da página pode ser visto no apêndice A.

Ao realizar a programação da mensagem foi utilizado no computador o servidor *apache XAMPP*. Esse servidor possibilita a comunicação através de portas entre o roteador que o computador está conectado e o microcontrolador que possui a placa *ethernet*.

Para ter acesso a página onde poderá monitorar e controlar sua residência, o usuário deverá entrar dados como *login* (autenticação) e senha, forma pela qual o usuário poderá restringir o acesso somente às pessoas autorizadas que desejam utilizar o controle de sua moradia, conforme próxima figura.



The image shows a web-based login form. At the top, it says "Por favor, coloque o login e senha para acessar a página". Below this, there are two input fields: "Login:" with the text "andre" entered, and "Senha:" with four dots "...." entered. A "Submit" button is located below the password field. At the bottom of the form, it says "Powered by Password Protect".

FIGURA 18 - CONTROLE DE ACESSO
FONTE: O AUTOR (2014).

Para realização desta autenticação foi utilizado uma biblioteca pronta em php chamada Page Password Protect 2.13.

Após feito este processo, uma página será aberta conforme Figura 19.



FIGURA 19 - PÁGINA DA CASA INTELIGENTE
FONTE: O AUTOR (2014).

Nesta página o usuário poderá visualizar a planta da sua residência e à partir dela escolher qual o ambiente que deseja fazer o controle e monitoramento.

Quando o usuário acessa algum ambiente uma nova página se abre, esta nova página (figura 20) mostrará informações como:

- Equipamentos que estão sendo monitorados que podem ser controlados;
- Sensores de temperatura, umidade;
- Lâmpadas.

Para a página web se conectar com o módulo ethernet acoplado no microcontrolador foi necessário definir o mesmo ip e mesma porta na página web.

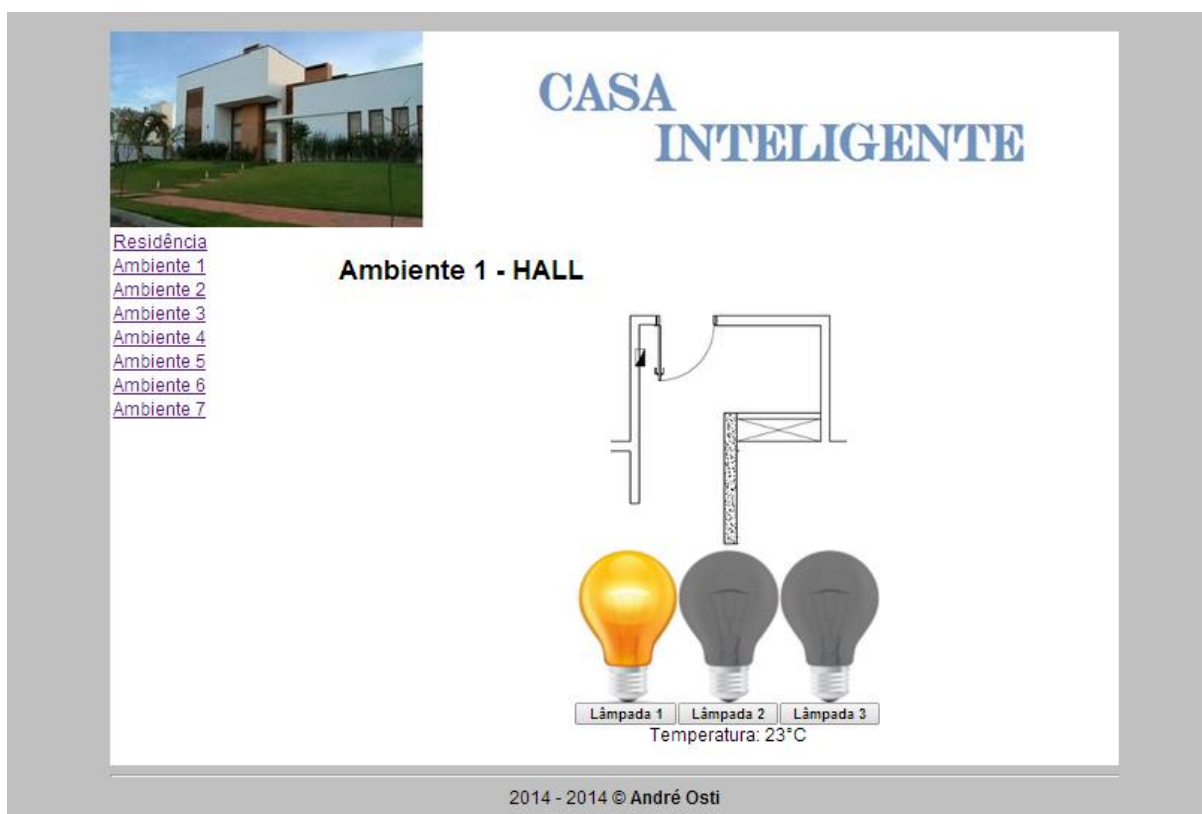


FIGURA 20 - PÁGINA REFERENTE AO AMBIENTE 1
 FONTE: O AUTOR (2014).

3.7 PROGRAMAÇÃO

3.7.1 Introdução

A comunicação serial é usada para programação dos chips ATmega328, já que estes já possuem *bootloader* de fábrica, ou seja, o sistema operacional já vem carregado e pronto para uso.

A programação foi feita utilizando o *software*(programa) Arduino IDE, de código aberto. O ambiente Arduino é escrito em java e baseado no programa Processing, onde uma das possíveis linguagens de programação utilizadas é o C, esta foi escolhida para o projeto.

A programação realizada se divide em três tipos:

- Programação do supervisor web;
- Programação para a central;
- Programação para os módulos.

Para o uso dos dispositivos de comunicação foram utilizados diversos tipos de bibliotecas prontas a fim de garantir o correto envio e recebimento das mensagens, porém as mensagens que serão enviadas de um módulo para outro ou de um módulo para a central e vice-versa foi criada pelo autor deste trabalho seguindo regras que serão melhor explicadas a seguir.

O principal componente da programação é a declaração de variáveis de cada ambiente:

```
char Luz1[7] = "0001L#";
```

Esta variável é uma *string*, ou seja, uma cadeia de caracteres que definirá o estado de cada equipamento: ligado (1) ou desligado (0). Os três primeiros dígitos representam três equipamentos e seus estados: 1 ou 0. O quarto dígito representa o ambiente que está sendo controlado, no caso mostrado seria o primeiro ambiente. O quinto dígito representa que se deseja controlar os equipamentos ou que se deseja saber o estado desses equipamentos, que será melhor explicado nos próximos tópicos. E o último dígito é a interpretação que um comando foi recebido e sempre e terá o símbolo "#".

A seguir será exemplificado uma situação do que ocorre no sistema no envio das mensagens. Para o exemplo será enviado via internet o comando para acender duas lâmpadas do primeiro ambiente e uma lâmpada do segundo ambiente.

Primeiramente os módulos enviam o estado de cada ambiente para a central de forma automática em intervalos constantes de tempo.

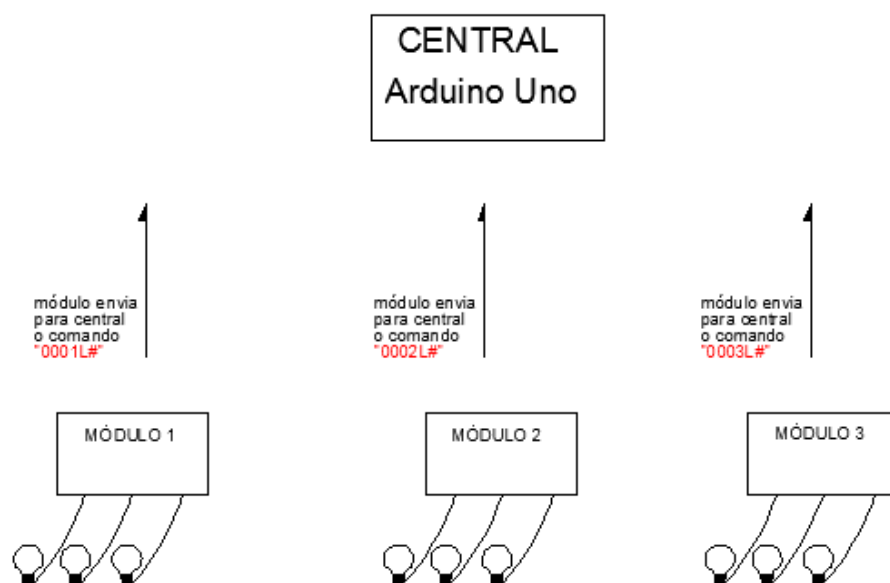


FIGURA 21 – ESQUEMÁTICO 1
FONTE: O AUTOR (2014).

Após a central receber o estado de cada ambiente e armazenar em sua memória o computador ou dispositivo móvel que está conectado à internet solicita esses estados dos ambientes para a central e ela retorna estes.



FIGURA 22 – ESQUEMÁTICO 2
 FONTE: O AUTOR (2014).

Quando o usuário deseja acender as lâmpadas de diferentes ambientes, ele envia através de seu dispositivo, via internet, o comando de acendimento para a central.

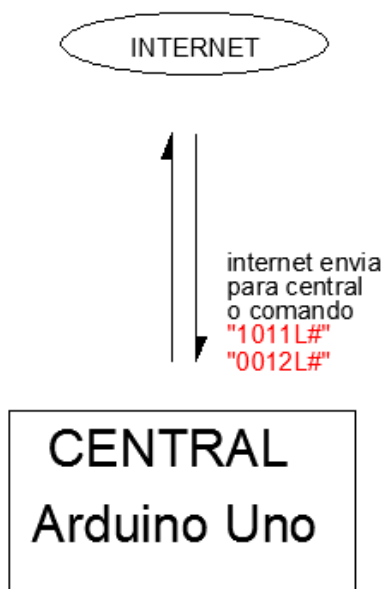


FIGURA 23 – ESQUEMÁTICO 3
 FONTE: O AUTOR (2014).

A central analisa a mensagem que foi recebida e envia para o módulo, este também analisa a mensagem e acende as lâmpadas. Após o acendimento o módulo retorna a central o novo estado do sistema, assim a central sempre saberá o estado de cada módulo que representa cada ambiente.

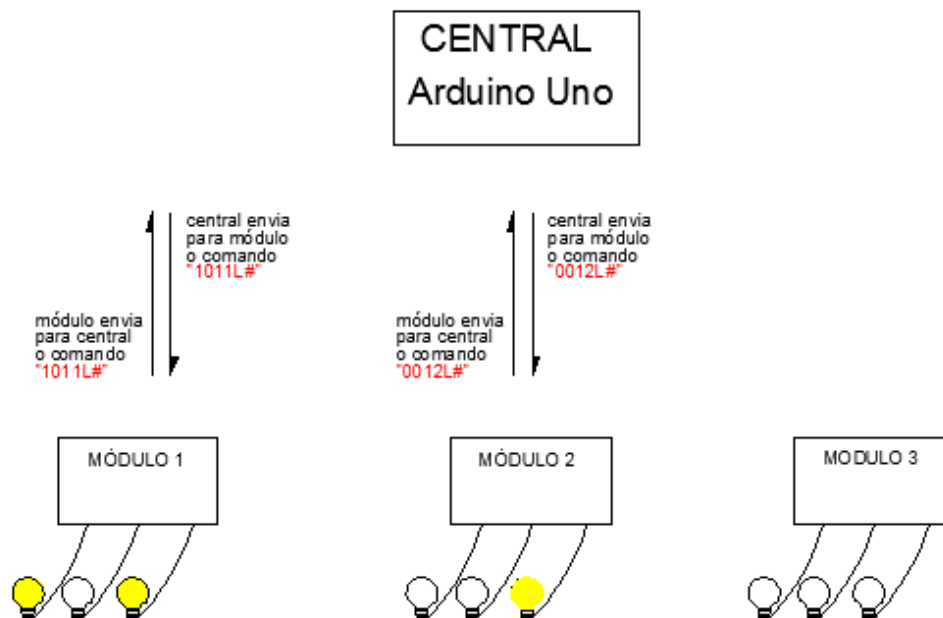


FIGURA 24 – ESQUEMÁTICO 4
FONTE: O AUTOR (2014).

Este processo irá acontecer enquanto o sistema estiver ligado. Caso o usuário realiza um comando via *bluetooth* ou infravermelho o comando será sempre enviado a central para ela estar sempre atualizada em relação ao sistema todo.

Com a definição do endereçamento de cada mensagem, para a comunicação ocorrer corretamente foram utilizadas bibliotecas prontas, ou seja, foi usado um conjunto de funções pré-escritas por outros programadores. A Tabela 3 exemplifica as bibliotecas utilizadas.

TABELA 3 – BIBLIOTECAS

Dispositivo	HARDWARE	BIBLIOTECA
RF433	RF 433MHz	VirtualWire
RS485	MAX485	VirtualWire
RF2,4Ghz	nRF24L01	NRF24
SUPERVISÓRIO/Ethernet	W5100	SPI, servidor XAMPP(apache)
BLUETOOTH	BLUETOOTH	O autor
INFRVERMELHO	TSOP4838	IRremote

FONTE: O AUTOR, 2014.

3.7.2 Programação do supervisor web

As páginas dos ambientes variam suas características conforme os equipamentos nelas instalados. Cada projeto de casa inteligente possuirá páginas que variam conforme os equipamentos de cada usuário.

Para a página de cada ambiente do supervisor foi definido caracteres diferentes para declaração da variável Luz anteriormente citada. Pode-se notar a seguir que a variável Luz agora é tratada como msg, pois representa uma mensagem que está sendo enviada para o arduino central.

```
$msg = "0001L#";
```

O quarto dígito representa o ambiente em que se deseja controlar, podendo ser de 1 a 7 respectivamente. A página da web necessita saber o estado em que cada equipamento se encontra e para isso ela solicita o estado de cada equipamento para a central através da mudança do quinto dígito (A,B,C,D,E,F,G):

```
$msg = "0001A#"; //receber estado dos equipamentos do ambiente 1
```

Quando o usuário deseja controlar sua residência pelo supervisor, ao acionar algum equipamento através dos botões encontrados no site o quinto dígito muda para L, enviando assim o novo estado dos equipamentos escolhidos:

```
$msg = "0101L#";
```

No exemplo("0101L#") nota-se que está sendo enviado para o microcontrolador("0101L#") o acionamento do segundo equipamento("0101L#") do primeiro ambiente("0101L#").

Finalmente, para requisitar os valores dos sensores de temperatura e umidade a página da web modifica o quinto caractere para U ou T, representando a umidade e temperatura respectivamente.

```
$msg = "U#"; // requisição de temperatura
```

```
$msg = "T#"; //requisição de umidade
```

3.7.3 Programação da central

A seguir, será mostrada as principais linhas de programação para a central. Para acesso a programação completa visualizar o apêndice.

O primeiro passo para programação do microcontrolador é a definição de variáveis e também de bibliotecas para os módulos de comunicação:

```
#include <SPI.h> //biblioteca spi
#include <Ethernet.h> //biblioteca para módulo ethernet
#include <VirtualWire.h> //biblioteca virtualwire
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //ethernet shield
byte ip[] = { 192,168,0, 99 }; //ip escolhido para o microcontrolador
byte gateway[] = { 192,168,0, 1 }; //ip do roteador
byte subnet[] = { 255, 255, 255, 0 }; //máscara de rede
EthernetServer server(8081); //porta escolhida para o servidor
char Luz1[7] = "0001L#"; //string do estado do primeiro ambiente
char Luz2[7] = "0002L#"; //string do estado do segundo ambiente
char Luz3[7] = "0003L#"; //string do estado do terceiro ambiente
char Luz4[7] = "0004L#"; //string do estado do quarto ambiente
char Luz5[7] = "0005L#"; //string do estado do quinto ambiente
char Luz6[7] = "0006L#"; //string do estado do sexto ambiente
char Luz7[7] = "0007L#"; //string do estado do sétimo ambiente
char msg[7] = "0000L#"; // String onde é guardada as msgs recebidas da web
```

O microcontrolador ficará em constante processo de recebimento de comandos vindos do supervisório e também dos módulos. Primeiramente o programa analisará o estado de cada ambiente vindo dos módulos e em seguida a conexão com a internet.

Caso haja conexão com a internet o microcontrolador fará a leitura do que está recebendo do supervisório através do comando:

```
msg[1]=msg[2];msg[2]=msg[3];msg[3]=msg[4];msg[4]=msg[5];
msg[5]=msg[6];
msg[6] = client.read();
```

A função `cliente.read()` faz a leitura pelo módulo ethernet do que for enviado pelo supervisório e assim é armazenado em uma string chamada “msg”.

Após o armazenamento o microcontrolador passa a analisar cada caractere desta mensagem em sequência.

Por exemplo:

```

if (msg[6]=='#') // analisa se o último caractere é o #
{
    if (msg[5]=='A') client.write(Luz1);
    if (msg[5]=='L')
    {
        if (msg[4]=='1')
        {
            // Caso L#, ele copia os bytes anteriores p/ a
            // string 'Luz' e cada byte representa um
            // dispositivo, onde '1'=ON e '0'=OFF
            Luz1[0]=msg[1];
            Luz1[1]=msg[2];
            Luz1[2]=msg[3];
            if (Luz1[0]=='1') comando1(); else comando2();
            if (Luz1[1]=='1') comando3(); else comando4();
            if (Luz1[2]=='1') comando5(); else comando6();
        }
    }
}

```

Sempre será analisado os dois últimos caracteres da mensagem. Caso o último for “#” quer dizer que foi recebido um comando. Ao analisar o penúltimo caractere ocorrerá duas opções, ou será um caractere requisitando o estado dos equipamentos em relação a um determinado ambiente (A,B,C,D,E,F,G), ou será um caractere(L) que manda executar o acionamento dos equipamentos.

Quando receber uma mensagem de acionamento (liga, desliga) o ATmega328 executa a função comando(). Esta função representa o que deve ser enviado via rádio frequência ou RS485 para os demais módulos conforme tabela3.

TABELA 4 – CÓDIGO ENVIADO PARA OS MÓDULOS

Ambiente	Liga Equip. 1	Desliga Equip. 1	Liga Equip. 2	Desliga Equip. 2	Liga Equip. 3	Desliga Equip. 3
1	010	011	012	013	014	015
2	020	021	022	023	024	025
3	030	031	032	033	034	035
4	040	041	042	043	044	045
5	050	051	052	053	054	055
6	060	061	062	063	064	065
7	070	071	072	073	074	075

FONTE: O AUTOR, 2014.

O primeiro dígito (0) representa a central, o segundo representa o ambiente de 1 a 7 e o terceiro representa o que deve ser feito com o equipamento, ligar ou desligar.

O envio desses códigos é feito pela seguinte programação:

```
void comando1()
{
    char *msg = "010";
    vw_send((uint8_t *)msg, strlen(msg));
}
```

3.7.4 Programação dos módulos

Os módulos possuem o papel de leitura dos sensores e acionamento dos equipamentos nos ambientes que estão acoplados.

Para realizar o acionamento dos equipamentos os módulos analisam se estão recebendo alguma mensagem através do código:

```
uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = VW_MAX_MESSAGE_LEN;
```

Após recebida uma mensagem, esta mensagem é analisada seguindo a tabela 1 anteriormente mostrada.

```
if((buf[0] == '0') && (buf[1] == '1')&&(buf[2] == '0'))
```

```
{  
    digitalWrite(saida1,1);  
}
```

Se a leitura da mensagem for igual a um valor existente na tabela, então o microcontrolador mudará o estado de um pino de saída digital pré-configurada. Este pino de saída digital acionará o relé que for configurado.

O módulo também receberá comandos feitos por bluetooth e infravermelho. Esses comandos serão interpretados e conforme os códigos recebidos será executada uma ação nos equipamentos do ambiente.

No final da execução desses comandos o controlador envia à central o novo estado do ambiente. Com esse envio de estado, caso a central perca a sua fonte de energia, ela receberá novamente o comportamento de cada ambiente, não perdendo assim o controle do sistema.

4 PROTÓTIPO

4.1 PRIMEIRA ETAPA

A partir do esquemático mostrado na figura 2 foi desenvolvido um protótipo inicial a fim de observar o comportamento do sistema utilizando os microcontroladores e os dispositivos que realizam as comunicações.

Primeiramente a central composta pelo Arduino, ethernet shield, rádio transmissor e receptor 433Mhz, módulo MAX485 e rádio nRF24L01 foi conectada da maneira apresentada na figura 25.

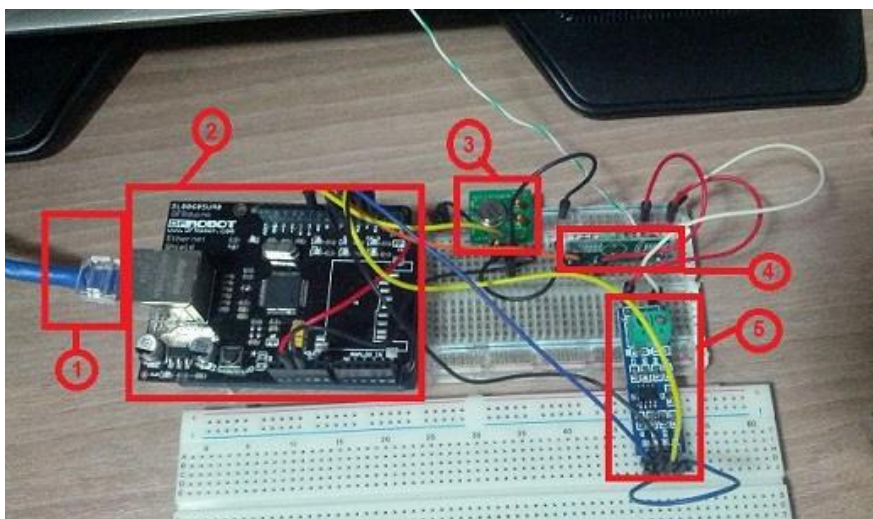


FIGURA 25 – CENTRAL

FONTE: O AUTOR (2014).

TABELA 5 – SIMBOLOGIA CENTRAL

Número	Componente
1	Cabo ethernet com conexão RJ45
2	Ethernet shield acoplado a um Arduino
3	Módulo RF433Mhz
4	nRF24L01
5	MAX485 para comunicação RS485

FONTE: O AUTOR, 2014.

Após feita a montagem da central, montou-se os módulos conforme a figura 26.

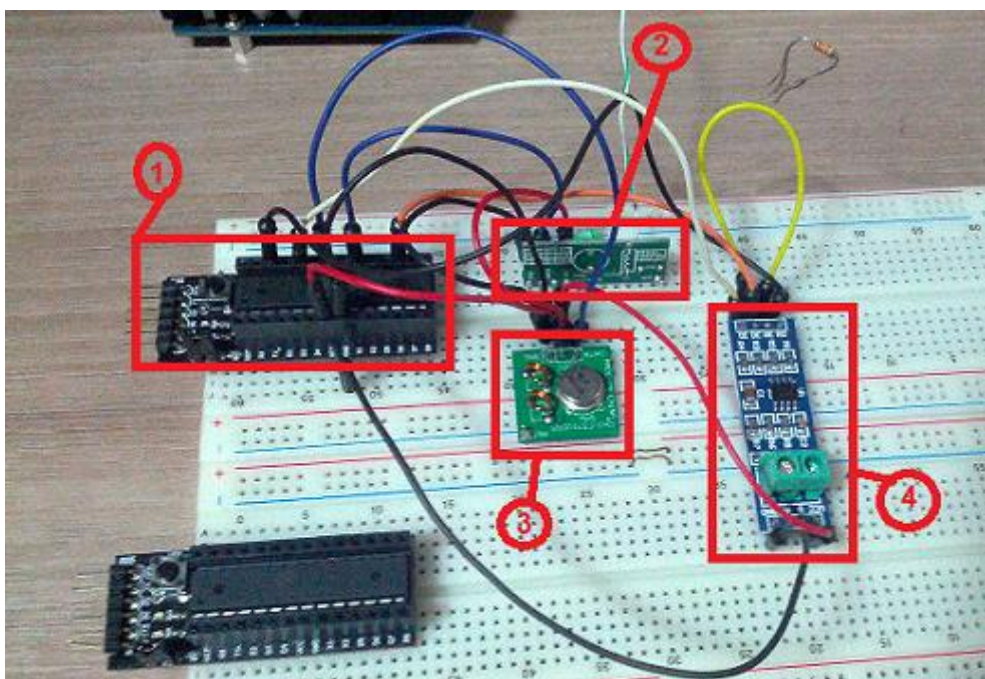


FIGURA 26 – MÓDULO

FONTE: O AUTOR (2014).

TABELA 6 – SIMBOLOGIA MÓDULO

Número	Componente
1	ATmega328
2	nRF24L01
3	Módulo RF433Mhz
4	MAX485 para comunicação RS485

FONTE: O AUTOR, 2014.

Esta montagem serviu de base para teste de comportamento das mensagens enviadas entre cada microcontrolador no sistema assim como as conexões necessárias para o correto funcionamento de cada dispositivo.

4.2 SEGUNDA ETAPA

A segunda etapa a ser realizada foi de criação de esquemático com todas as ligações necessárias da central e dos módulos. Estes foram divididos em dois tipos: com o microcontrolador atmega328 e o outro com o microcontrolador attiny85.

Para realização do esquemático foi utilizado o programa Eagle. Este programa tem a função de conectar os componentes utilizados e apartir do esquemático criar uma placa de circuito impresso.

Para o módulo microcontrolador por atmega328 foi criado o esquemático conforme Apêndice F.

Com o esquemático pronto, foi possível criar uma placa de circuito impresso conforme Figura 27.

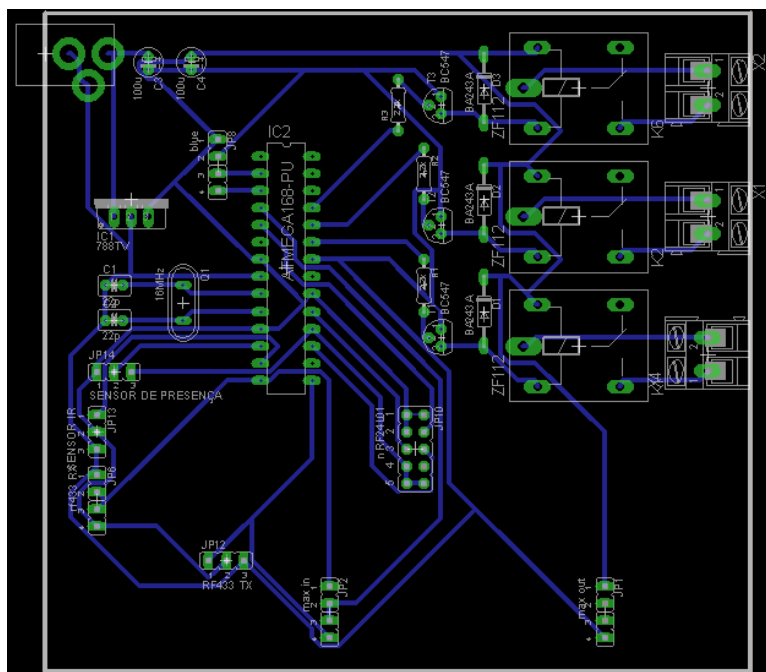


FIGURA 27 – PCB MÓDULO ATMEGA328
FONTE: O AUTOR (2014).



FIGURA 28 – PLACA MÓDULO ATMEGA328
FONTE: O AUTOR (2014).

O módulo da figura 28 possui três saídas a relé com contato normalmente aberto para acionamento de lâmpadas, ventiladores, entre outros, podendo operar em uma faixa de até 15A e 125Vac ou 24Vcc. Este módulo encontra-se pronto para ser controlado por infra-vermelho via controle remoto, *bluetooth* via dispositivo móvel, assim como pelo supervisor conectado à internet. Este supervisor envia sinais para acionamento via rádio frequência e RS485. Pode-se conectar uma fonte de 5Vcc até 30Vcc para alimentação do módulo. Também foi projetado pinos para conexão de sensor de presença. Estes se encontram ao lado direito do Crystal de 16Mhz.

Já o módulo com o microcontrolador attiny85 possui o esquemático e placa de circuito impresso conforme as ilustrações seguintes.

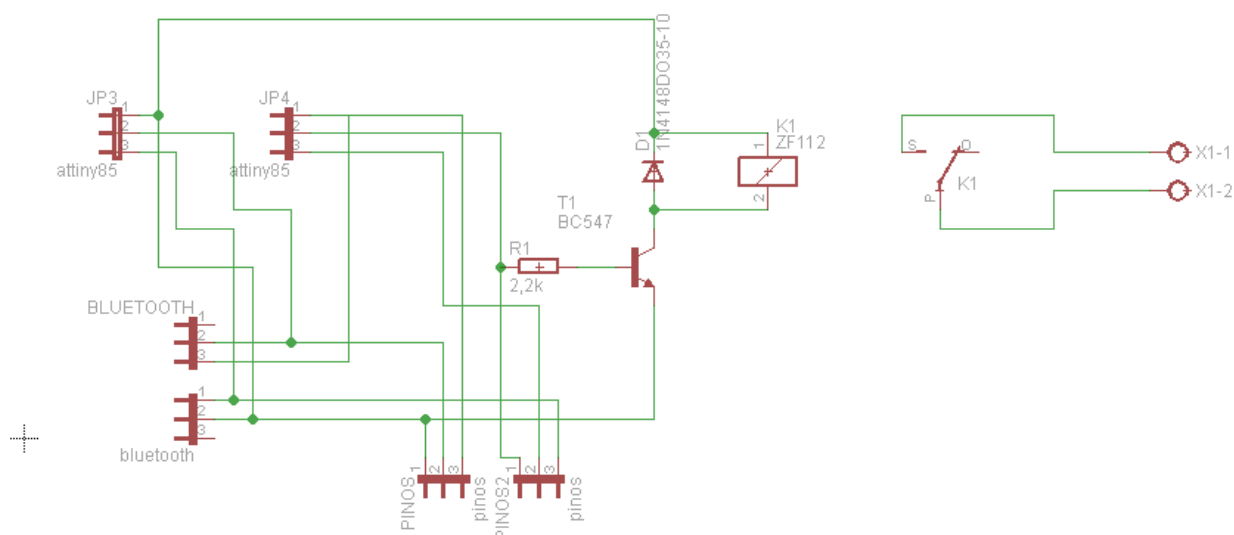


FIGURA 29 – ESQUEMÁTICO MÓDULO ATTINY85
FONTE: O AUTOR (2014).

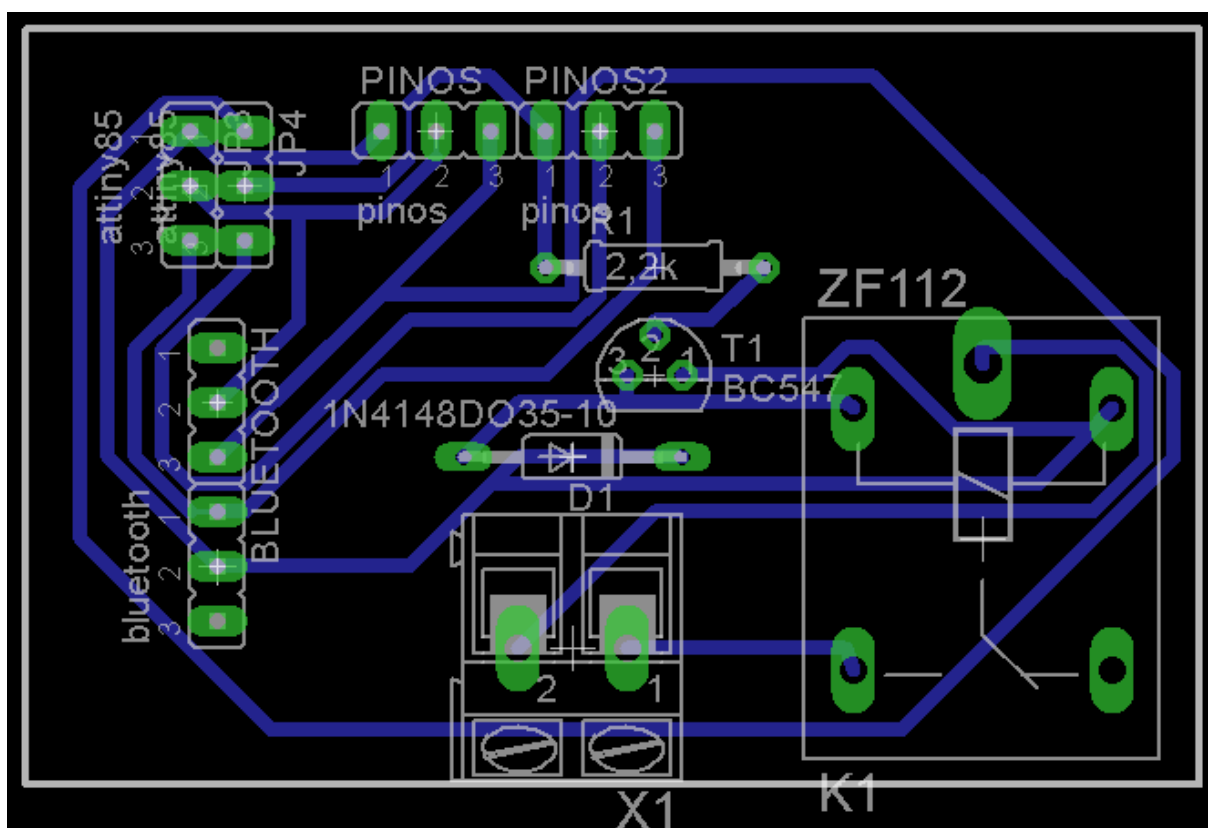


FIGURA 30 – PCB MÓDULO ATTINY85
FONTE: O AUTOR (2014).

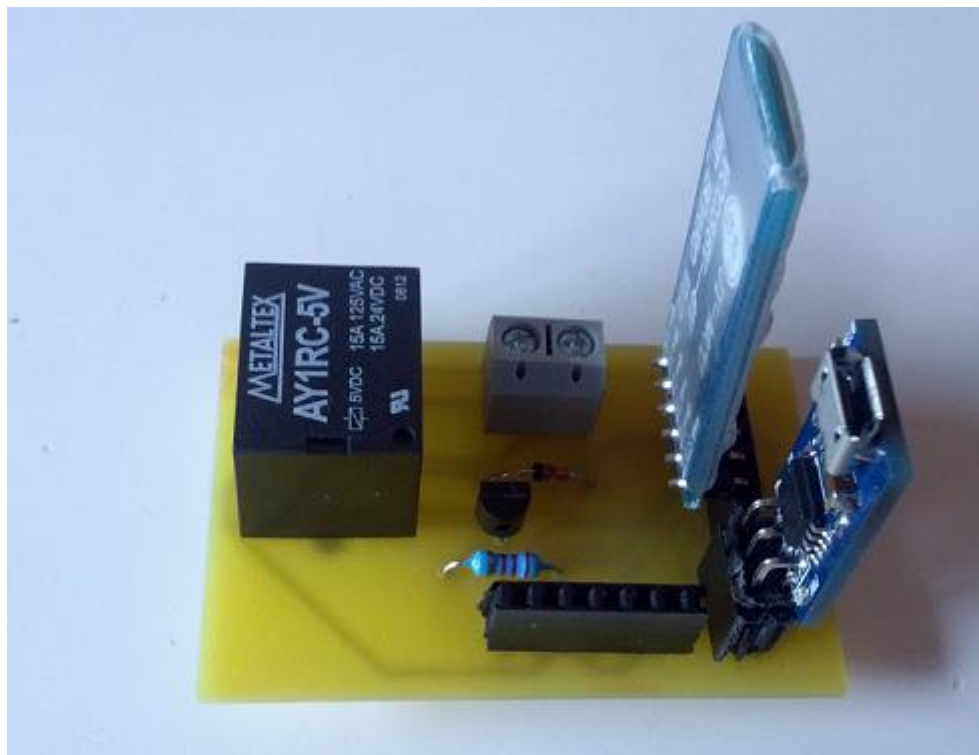


FIGURA 31 – PLACA MÓDULO ATTINY85
FONTE: O AUTOR (2014).

Este módulo é capaz de se comunicar via *bluetooth* com um dispositivo móvel com sistema operacional Android. Foram acoplados pinos ao lado do microcontrolador para expansões futuras de uso das portas digitais, podendo assim ser programadas e usadas com sensores. A placa criada possui uma saída a relé conforme especificações dadas no módulo anterior.

Caso o usuário não possua o aplicativo que foi criado para demonstração no laboratório de Engenharia Elétrica, este poderá controlar utilizando um aplicativo que envie dados para o módulo *bluetooth*. Para conexão com o módulo *bluetooth* basta o usuário parear seu dispositivo móvel com o dispositivo 'linvor', nome dado ao módulo *bluetooth*. Ao fazer o pareamento será solicitado uma senha para conexão, cuja senha a ser digitada é: 1234.

Para o módulo com ATmega328 as saídas fecharão o contato ao receberem caracteres 'D', 'E', 'F' e abrirão os contatos ao receberem os caracteres 'a', 'b', 'c'. No caso do módulo com ATtiny85 a saída fechará o contato ao receber o caractere 'A' e abrirá o contato ao receber o caractere 'a'.

4.3 TERCEIRA ETAPA

A terceira etapa compreende a montagem do sistema de forma completa, ou seja, com a central, os módulos e equipamentos a serem acionados.

Para esta etapa os módulos e a central foram colocados em caixas de madeira MDF.

O protótipo final foi montado em uma base de madeira para, de forma didática, testar todos os equipamentos e comunicações disponíveis no acionamento de lâmpadas.

A seguir será mostrado na Tabela 7 todos os itens utilizados no projeto para implementação da central, módulos e do supervisor para em trabalhos futuros ser feito o desenvolvimento a partir destes itens. Para realização da programação dos supervisórios e o uso do *bluetooth*, o autor deste projeto foi responsável por criar as lógicas e portanto podem ser vistas na Tabela 7 onde se encontram essas lógicas no trabalho.

TABELA 7 – LISTA DE ITENS UTILIZADOS

ITEM	HARDWARE	SOFTWARE	ONDE OBTER
1	ATmega328	-	http://www.labdegaragem.org/loja/garagino-4/garagino-rev1.html
2	Shield ethernet	-	http://www.labdegaragem.org/loja/ethernet-shield.html
3	RF433	-	http://imall.iteadstudio.com/im120628014.html
4	MAX485	-	http://imall.iteadstudio.com/max485-module.html
5	nRF24L01	-	http://imall.iteadstudio.com/im120606002.html
6	-	Supervisório/ Programação	Capítulo 3.7
7	-	Página web	Apêndice A e B
8	-	Arduino IDE	www.arduino.cc
9	-	VirtualWire	http://www.airspayce.com/mikem/arduino/VirtualWire/
10	-	IRremote	http://arcfn.com
11	-	RF24	http://maniacbug.github.io/RF24/
12	Attiny85	-	http://imall.iteadstudio.com/im130615003.html
13	Relé, resistores, diodos, transistores	-	Loja de matérias eletrônicas
14	Bluetooth	-	http://imall.iteadstudio.com/im120723009.html
15	-	Bluetooth	Apêndice D

FONTE: O AUTOR, 2014.

5 CONCLUSÃO

5.1 CONSIDERAÇÕES FINAIS

Desenvolver um sistema que possa além de ser funcional, mas também economicamente acessível, pode trazer a sociedade a maior difusão da automação residencial no Brasil. Os brasileiros já possuem em suas residências equipamentos que podem ser automatizados e estão apenas a um pequeno passo de aumentar ainda mais o conforto, segurança e qualidade de vida.

A automação residencial já está deixando de fazer parte somente do cotidiano de pessoas de maior poder aquisitivo no Brasil. Com o avanço da tecnologia e com a implementação de sistemas cada vez mais baratos mostra que populações de classe média e também de baixa renda têm a oportunidade de automatizar seus domicílios.

Com esse trabalho pôde-se mostrar detalhadamente como pode ser projetado e implementado um sistema de monitoramento e controle de uma residência com diversos tipos de comunicação. O protótipo desenvolvido mostrou sua *performance* de forma robusta e satisfatória nos ensaios realizados no laboratório do departamento de Engenharia Elétrica da Universidade Federal do Paraná.

5.2 TRABALHOS FUTUROS

Como melhoria futura para criar um sistema com melhores características de processamento, mais funcionalidade, pode-se estudar:

- O uso de uma central mais elaborada (raspberry pi, mini pc)
- Inserir uma central com LCD sensível ao toque
- Inserir sistema de circuito fechado de tv (CFTV)
- Rede mais inteligente com Zigbee
- Inserir criptografia entre as comunicações.
- Adequação do sistema em protocolos já existentes através de um circuito adaptador.

- Extensão para edificações como shoppings e condomínios.
- Interoperabilidade independente de fabricantes
- Flexibilidade e expansão (com adicionamento de pontos de entradas/saídas-I/O).

REFERÊNCIAS

ABNT (s.d.). **Associação Brasileira de Normas Técnicas**. Disponível em: <<http://www.abnt.org.br/>>. Acesso em 10 de agosto de 2013.

ACCARDI, A.; DODONOV, E. (2012). Tecnologias, Infraestrutura e Software, **Automação residencial: Elementos básicos, arquiteturas, setores, aplicações e protocolos**, São Carlos, v. 1, n. 2, p.156-166, nov. 2012.

ADAFRUIT (2011). **Biblioteca DHT22**. Disponível em: <<https://github.com/adafruit/DHT-sensor-library>>. Acesso em 25 de maio de 2014.

ARDUINO. **Kit de desenvolvimento Arduino**. Disponível em: <<http://www.arduino.cc>>. Acesso em 8 de agosto de 2013.

ATMEL (2013). **Especificações Técnicas ATtiny85**. Disponível em: <http://www.atmel.com/images/atmel-2586-avr-8-bit-microcontroller-attiny25-attiny45-attiny85_datasheet.pdf>. Acesso em 05 de novembro de 2014.

ATMEL (s.d.). **Especificações Técnicas ATmega328**. Disponível em: <<http://www.atmel.com/Images/doc8161.pdf>>. Acesso em 05 de novembro de 2014.

AURESIDE (s.d.). **Associação Brasileira de Automação residencial**. Disponível em: <<http://www.aureside.org.br/default.asp>>. Acesso em: 20 de Novembro de 2013.

AURESIDE (2014). **Mercado brasileiro de Automação é destaque em publicação alemã**. Disponível em: <http://www.aureside.org.br/noticias_recentes/default.asp?file=01.asp&id=354>. Acesso em: 17 de março de 2014.

BOLZANI, C. A. M. (2011). **Smart Homes And Support to those with Alzheimer`s**. Disponível em: <<http://www.bolzani.com.br/2011/03/smart-homes-support-alzheimer/>>. Acesso em: 15 de Novembro de 2013.

BOLZANI, C. A. M. (2004). **Residências Inteligentes**, Ed. Livraria da Física, São Paulo, Brasil.

BOLZANI, C. A. M. (2011). **Residências inteligentes e o novo contexto do século XXI**. Disponível em: <<http://www.bolzani.com.br/2011/12/residencias-inteligentes-e-o-novo-contexto-do-seculo-xxi/>>. Acesso em: 15 de novembro de 2013.

BONATTO, A.; Canto, D. O. (2007). **BLUETOOTH TECHNOLOGY (IEEE 802.15)**. Artigo Faculdade de Administração, Contábeis e Economia, PUCRS.

CEDOM (2013). **Associação Espanhola de Domótica**. Disponível em: <<http://www.cedom.es/sobre-domotica/que-es-domotica>>. Acesso em: 20 de novembro de 2013.

CISCO (2013). **Internert das coisas**. Disponível em: <<http://internetofeverything.cisco.com/pt-br>>. Acesso em: 20 de novembro de 2013.

DIAS, C. L. A; PIZZOLATO (2004). **DOMÓTICA, Aplicabilidade e Sistemas de Automação Residencial** (2004). Ed. Essentia Editora, v.6, n.3 (2004) Rio de Janeiro, Brasil.

FERREIRA, E. H. C (2009). **Automação Residencial Utilizando Protocolo Can**. Trabalho de Conclusão de Curso de Engenharia Elétrica da Universidade Federal do Paraná. Disponível em: <<http://www.eletrica.ufpr.br/p/arquivostccs/28.pdf>>. Acesso em: 24 de março de 2014.

HAYKIN S.; MOHER M (2011). **Sistemas de Comunicações** (5 ed.). Editora Bookman.

HERNASKI, S. R. (2010). **Domótica Automação Residencial**. Trabalho de Conclusão de Curso de Engenharia Elétrica da Universidade Federal do Paraná. Disponível em: <<http://www.eletrica.ufpr.br/p/arquivostccs/145.pdf>>. Acesso em: 25 de março de 2014.

IBGE (2010). **Resultados do Universo do Censo Demográfico 2010**. Disponível em: <<http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2011/default.shtm>>. Acesso em: 10 de março de 2014.

MCCAULEY, M. **Biblioteca VirtualWire**. Disponível em: <<http://www.airspayce.com/mikem/arduino/VirtualWire/>>. Acesso em: 07 de agosto de 2013.

MENDES, M. B.; MARCENGO, T. M. (2011). **Domótica voltada para plataformas móveis com sistema operacional android**. Trabalho de Conclusão de Curso de Engenharia Elétrica da Universidade Federal do Paraná. Disponível em: <<http://www.eletrica.ufpr.br/p/arquivostccs/194.pdf>>. Acesso em: 17 de janeiro de 2014.

MURATORI, J. R.; DAL BÓ, P. H (s.d.). **Automação residencial: histórico, definições e conceitos**. Disponível em: <<http://www.osetoreletrico.com.br/web/component/content/article/70-automacao-residencial/590-capitulo-i-automacao-residencial-historico-definicoes-e-conceitos.html>>. Acesso em: 18 de março de 2014.

NICKEL, E. D.; BESSA W. K. S. M. (2010). **Sistema embarcado com acesso sem-fio**. Trabalho de Conclusão de Curso de Engenharia Elétrica da Universidade Federal do Paraná. Disponível em: <<http://www.eletrica.ufpr.br/p/arquivostccs/157.pdf>>. Acesso em: 18 de janeiro de 2014.

NORDIC (2007). **Datasheet nRF24L01**. Disponível em: <ftp://imall.iteadstudio.com/Modules/IM120606002_nRF24L01_module/DS_nRF24L01.pdf>. Acesso em: 20 de novembro de 2013.

OLHAR DIGITAL (2014). **Internet das Coisas**. Disponível em: <<http://olhardigital.uol.com.br/pro/noticia/40638/40638>>. Acesso em: 17 de Março de 2014.

SHIRRIFF, K. (2009). **Biblioteca IRremote**. Disponível em: <<http://arcfn.com>>. Acesso em: 11 de janeiro de 2014.

SILVA, D. S. (2009), **Desenvolvimento e Implementação de um Sistema de Supervisão e Controle Residencial**. Disponível em: <<ftp://ftp.ufrn.br/pub/biblioteca/ext/bdtd/DaniseSS.pdf>>. Acesso em 22 de novembro de 2013.

SIPEX (2002). **Datasheet MAX485**. Disponível em: <<https://www.sparkfun.com/datasheets/Components/General/sp3485CN-LTR.pdf>>. Acesso em 25 de maio de 2014.

TEIXEIRA, W. (2011). **O que é DALI?** Disponível em:
<http://www.lumearquitetura.com.br/pdf/ed21/ed_21_Aula.pdf>. Acesso em 20 de maio de 2014.

TEXAS INSTRUMENTS (1999). **Datasheet LM35**. Disponível em:
<<http://www.ti.com/lit/ds/symlink/lm35.pdf>>. Acesso em 23 de fevereiro de 2014.

THOMAZINI, D. e ALBUQUERQUE, P., **Sensores Industriais – Fundamentos e aplicações**, 4ª. Edição, Editora Érica, 2005.

WIZNET (2011). **Datasheet W5100**. Disponível em:
<http://www.wiznet.co.kr/UpLoad_Files/ReferenceFiles/W5100_Datasheet_v1.2.4.pdf>. Acesso em 18 de agosto de 2013.

APÊNDICE A – PÁGINA HOME

```

<html>
<php>
<?php include("password_protect.php"); ?>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="style.ccs" />
<title>
In&#237cio - Automa&#231&#227o Residencial
</title>
<style type="text/css">
<!--
#logoApDiv {
position:absolute;
width:900px;
height:175px;
z-index:1;
top: 20px;
left: 50%; /*posição em relação à parte esquerda*/
margin-left: -450px;
border:1px;
background-color: #FFFFFF;
//background:url(gradial.png);
}
#menuApDiv {
position:absolute;
width:250px;
height:100%;
z-index:2;
top: 195px;
left: 50%; /*posição em relação à parte esquerda */
margin-left: -450px;
border:1px;
background-color: #FFFFFF;
}
#ContApDiv {
position:absolute;
width:650px;
z-index:3;
left: 50%; /*posição em relação à parte esquerda */
margin-left: -200px;
top: 195px;
background-color: #FFFFFF;
}
#SiteApDiv {
position:absolute;
width:900px;

```

```

/* height -> none */
top:195px;
z-index:2;
left: 50%; /*posição em relação à parte esquerda */
margin-left: -450px;
}

}
-->
</style>
</head>
<body SCROLL="YES" background="back.jpg" >
<font face="Arial, Helvetica, sans-serif">
<div id="logoApDiv"><a href="/teste/home.php"></a>
</div>
<div id="SiteApDiv">
<table border="0px" width="900px" bgcolor="#FFFFFF"><!-- Tabela de layout -->
<tr>
<!-- Menu -->
<td width="200px" valign="top">
<table width="200" class="bordasimples">
<tr>
<td align="left" bgcolor="#FFFFFF" class="navlink"><a href="/teste/home.php">
Residência</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente1.php">Ambiente
1</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente2.php">Ambiente
2</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente3.php">Ambiente
3</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente4.php">Ambiente
4</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente5.php">Ambiente
5</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente6.php">Ambiente
6</a></td>
</tr>

```

```

<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente7.php">Ambiente
7</a></td>
</tr>
</table>
</td>
<!-- conteudo -->
<td width="700px">
<table>
<tr>
<td>
</td>
</tr>
</table><h1>Resid&#234ncia</h1>
<font size='2'><h2>Escolha o Ambiente em que deseja supervisionar:</h2></font>
<center>
<table>
<img src='planta.png' width='600' >
</table>
</center>
</td>
</tr>
</table>
<hr />
<center><font face="Arial, Helvetica, sans-serif">2014 - 2014 &copy; <b>Andr&#233
Osti</b></font></center>
</div>
</font>
</body>
</html>

```


APÊNDICE B – PÁGINA AMBIENTE 1

```

<html>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<php>
<?php include("password_protect.php"); ?>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="style.ccs" />
<title>
In&#237cio - Automa&#231&#227o Residencial
</title>
<style type="text/css">
<!--
#logoApDiv {
position:absolute;
width:900px;
height:175px;
z-index:1;
top: 20px;
left: 50%; /*posição em relação à parte esquerda*/
margin-left: -450px;
border:1px;
background-color: #FFFFFF;
//background:url(gradial.png);
}
#menuApDiv {
position:absolute;
width:250px;
height:100%;
z-index:2;
top: 195px;
left: 50%; /*posição em relação à parte esquerda */
margin-left: -450px;
border:1px;
background-color: #FFFFFF;
}
#ContApDiv {
position:absolute;
width:650px;
z-index:3;
left: 50%; /*posição em relação à parte esquerda */
margin-left: -200px;
top: 195px;
background-color: #FFFFFF;
}
#SiteApDiv {
position:absolute;

```

```

width:900px;
/* height -> none */
top:195px;
z-index:2;
left: 50%; /*posição em relação à parte esquerda */
margin-left: -450px;
}

}
-->
</style>
</head>
<body SCROLL="YES" background="back.jpg" >
<font face="Arial, Helvetica, sans-serif">
<div id="logoApDiv"><a href="/teste/home.php"></a>
</div>
<div id="SiteApDiv">
<table border="0px" width="900px" bgcolor="#FFFFFF"><!-- Tabela de layout -->
<tr>
<!-- Menu -->
<td width="200px" valign="top">
<table width="200" class="bordasimples">
<tr>
<td align="left" bgcolor="#FFFFFF" class="navlink"><a href="/teste/home.php">
Residência</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente1.php">Ambiente
1</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente2.php">Ambiente
2</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente3.php">Ambiente
3</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente4.php">Ambiente
4</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente5.php">Ambiente
5</a></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente6.php">Ambiente
6</a></td>

```

```

</tr>
<tr>
<td bgcolor="#FFFFFF" class="navlink"><a href="ambiente7.php">Ambiente
7</a></td>
</tr>
</table>
</td>
<!-- conteudo -->
<td width="700px">
<table>
<tr>
<td>
</td>
</tr>
</table><h2>Ambiente 1 - HALL</h2>
<center>
<img src='ambiente1.png'>
<?php
$sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
// Se conecta ao IP e Porta:
socket_connect($sock,"192.168.0.99", 8080);
// Executa a ação correspondente ao botão apertado.
if(isset($_POST['bits'])) {
    $msg = $_POST['bits'];
    if(isset($_POST['Lampada1'])) { if($msg[0]=='1') { $msg[0]='0'; } else { $msg[0]='1'; }}
    if(isset($_POST['Lampada2'])) { if($msg[1]=='1') { $msg[1]='0'; } else { $msg[1]='1'; }}
    if(isset($_POST['Lampada3'])) { if($msg[2]=='1') { $msg[2]='0'; } else { $msg[2]='1'; }}
    socket_write($sock,$msg,strlen($msg));
}
socket_write($sock,'A#',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]=='#')&&($status[3]=='1')) {
    if ($status[0]=='0') $img1 = 'lamp_off.png';
    else $img1 = 'lamp_on.png';
    if ($status[1]=='0') $img2 = 'lamp_off.png';
    else $img2 = 'lamp_on.png';
    if ($status[2]=='0') $img3 = 'lamp_off.png';
    else $img3 = 'lamp_on.png';
    echo "<form method =\"post\" action=\"ambiente1.php\">";
    echo "<input type=\"hidden\" name=\"bits\" value=\"\"$status\">";
    echo "<img src='$img1'>";
    echo "<img src='$img2'>";
    echo "<img src='$img3'><br>";
    echo "<button style=\"margin-right:1px;width:90;font: bold 12px Arial\" type =
'Submit' Name = 'Lampada1'>L&#226mpada 1</button>";
    echo "<button style=\"margin-right:1px;width:90;font: bold 12px Arial\" type =
'Submit' Name = 'Lampada2'>L&#226mpada 2</button>";
    echo "<button style=\"margin-right:1px;width:90;font: bold 12px Arial\" type =
'Submit' Name = 'Lampada3'>L&#226mpada 3</button>";

```

```

//Requisita o status de temperatura.
socket_write($sock,'T#',2);
$temp = socket_read($sock,2);
echo "</br>Temperatura: ";
echo $temp;
echo "°C";
}
// Caso ele não receba o status corretamente, avisa erro.
else { echo "Falha ao receber status da casa."; }
socket_close($sock);
?>
</td>
</tr>
</table>
<hr />
<center><font face="Arial, Helvetica, sans-serif">2014 - 2014 &copy; <b>Andr&#233
Osti</b></font></center>
</div>
</font>
</body>
</html>

```

APÊNDICE C – PROGRAMAÇÃO CENTRAL

```
//portas 4,10,11,12,13 nao podem ser usadas por causa da ethernet
```

```
#include <DHT.h> //biblioteca sensor de umidade
```

```
#define DHTPIN 2 // pino data na entrada digital 2
```

```
#define DHTTYPE DHT22 // DHT 22 (AM2302)
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include <VirtualWire.h>
```

```
//Configurações do Ethernet Shield
```

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

```
byte ip[] = { 192,168,0,99 }; // ip que o arduino assumirá
```

```
byte gateway[] = { 192,168,0,1}; // ip do roteador
```

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

```
EthernetServer server(8080); // Cria o servidor na porta 8080
```

```
// String que representa o estado dos dispositivos de cada ambiente
```

```
char Luz1[7] = "0001L#";
```

```
char Luz2[7] = "0002L#";
```

```
char Luz3[7] = "0003L#";
```

```
char Luz4[7] = "0004L#";
```

```
char Luz5[7] = "0005L#";
```

```
char Luz6[7] = "0006L#";
```

```
char Luz7[7] = "0007L#";
```

```
// String onde é guardada as msgs recebidas
```

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

```
void setup()
```

```
{
```

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

```
  server.begin();
```

```
  Serial.begin(9600);
```

```
  dht.begin();
```

```
  vw_set_tx_pin(9); // Define o pino do transmissor,o 9
```

```
  vw_set_rx_pin(8); // Define o pino receptor,o 8
```

```
  vw_set_ptt_pin(7);
```

```
  vw_setup(2000); // Bits per sec
```

```
}
```

```
void comando1() //comando de envio para os módulos
{
    char *msg = "010"; //mensagem que será enviada
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando2()
{
    char *msg = "011";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando3()
{
    char *msg = "012";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando4()
{
    char *msg = "013";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando5()
{
    char *msg = "014";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando6()
{
    char *msg = "015";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando7()
{
    char *msg = "020";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando8()
```

```

{
    char *msg = "021";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando9()
{
    char *msg = "022";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando10()
{
    char *msg = "023";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando11()
{
    char *msg = "024";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando12()
{
    char *msg = "025";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando13()
{
    char *msg = "030";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando14()
{
    char *msg = "031";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando15()
{

```

```

    char *msg = "032";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando16()
{
    char *msg = "033";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando17()
{
    char *msg = "034";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando18()
{
    char *msg = "035";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando19()
{
    char *msg = "040";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando20()
{
    char *msg = "041";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando21()
{
    char *msg = "042";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando22()
{
    char *msg = "043";

```



```

    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando23()
{
    char *msg = "044";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando24()
{
    char *msg = "045";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando25()
{
    char *msg = "050";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando26()
{
    char *msg = "051";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando27()
{
    char *msg = "052";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando28()
{
    char *msg = "053";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando29()
{
    char *msg = "054";
    vw_send((uint8_t *)msg, strlen(msg));

```

```

    vw_wait_tx(); // Wait until the whole message is gone
}
void comando30()
{
    char *msg = "055";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando31()
{
    char *msg = "060";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando32()
{
    char *msg = "061";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando33()
{
    char *msg = "062";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando34()
{
    char *msg = "063";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando35()
{
    char *msg = "064";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando36()
{
    char *msg = "065";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}

```

```

}
void comando37()
{
    char *msg = "070";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando38()
{
    char *msg = "071";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando39()
{
    char *msg = "072";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando40()
{
    char *msg = "073";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando41()
{
    char *msg = "074";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}
void comando42()
{
    char *msg = "075";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx(); // Wait until the whole message is gone
}

void loop()
{

    EthernetClient client = server.available(); //se houver conexão ethernet
    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]=='#')
  {
    if (msg[5]=='A') client.write(Luz1); //se for solicitado envia estado do
ambiente
    if (msg[5]=='B') client.write(Luz2);
    if (msg[5]=='C') client.write(Luz3);
    if (msg[5]=='D') client.write(Luz4);
    if (msg[5]=='E') client.write(Luz5);
    if (msg[5]=='F') client.write(Luz6);
    if (msg[5]=='G') client.write(Luz7);

    if (msg[5]=='U') //se for solicitado a umidade envia ela
    {
      float u = dht.readHumidity();
      client.print(u,2);
    }

    if (msg[5]=='T') //se for solicitado a temperatura envia ela
    {
      float t = dht.readTemperature();
      client.print(t,2);
    }

    if (msg[5]=='L') //se for solicitado o acendimento
    {
      if (msg[4]=='1')
      {
        // Caso L#, ele copia os bytes anteriores p/ a
        // string 'Luz' e cada byte representa um
        // dispositivo, onde '1'=ON e '0'=OFF
        Luz1[0]=msg[1];
        Luz1[1]=msg[2];
        Luz1[2]=msg[3];
        if (Luz1[0]=='1') comando1(); else comando2();
        if (Luz1[1]=='1') comando3(); else comando4();
        if (Luz1[2]=='1') comando5(); else comando6();
      }
    }
  }
}

```

```
if (msg[4]=='2')
{
Luz2[0]=msg[1];
Luz2[1]=msg[2];
Luz2[2]=msg[3];
if (Luz2[0]=='1') comando7(); else comando8();
if (Luz2[1]=='1') comando9(); else comando10();
if (Luz2[2]=='1') comando11(); else comando12();
}
```

```
if (msg[4]=='3')
{
Luz3[0]=msg[1];
Luz3[1]=msg[2];
Luz3[2]=msg[3];
if (Luz3[0]=='1') comando13(); else comando14();
if (Luz3[1]=='1') comando15(); else comando16();
if (Luz3[2]=='1') comando17(); else comando18();
}
```

```
if (msg[4]=='4')
{
Luz4[0]=msg[1];
Luz4[1]=msg[2];
Luz4[2]=msg[3];
if (Luz4[0]=='1') comando19(); else comando20();
if (Luz4[1]=='1') comando21(); else comando22();
if (Luz4[2]=='1') comando23(); else comando24();
}
```

```
if (msg[4]=='5')
{
Luz5[0]=msg[1];
Luz5[1]=msg[2];
Luz5[2]=msg[3];
if (Luz5[0]=='1') comando25(); else comando26();
if (Luz5[1]=='1') comando27(); else comando28();
if (Luz5[2]=='1') comando29(); else comando30();
}
```

```
if (msg[4]=='6')
{
```

```
Luz6[0]=msg[1];
Luz6[1]=msg[2];
Luz6[2]=msg[3];
if (Luz6[0]=='1') comando31(); else comando32();
if (Luz6[1]=='1') comando33(); else comando34();
if (Luz6[2]=='1') comando35(); else comando36();
}

if (msg[4]=='7')
{
Luz7[0]=msg[1];
Luz7[1]=msg[2];
Luz7[2]=msg[3];
if (Luz7[0]=='1') comando37(); else comando38();
if (Luz7[1]=='1') comando39(); else comando40();
if (Luz7[2]=='1') comando41(); else comando42();
}
}
}
}
```

APÊNDICE D – PROGRAMAÇÃO MÓDULO 1

```
#include <VirtualWire.h>
int estado1 = 0; //estado da saída
int estado2 = 0;
int estado3 = 0;
void setup()
{
    Serial.begin(9600);
    Serial.println("setup");
    pinMode (A0, OUTPUT);
    pinMode (A1, OUTPUT);
    pinMode (A2, OUTPUT);
    pinMode (3, OUTPUT);

    vw_set_rx_pin (8); // Define o pino do receptor
    vw_set_tx_pin(9); // Define o pino do transmissor
    vw_setup(2000); // Bits per sec
    vw_rx_start();
}

void loop()
{
    digitalWrite(3,HIGH);
    uint8_t buf[VW_MAX_MESSAGE_LEN];
    uint8_t buflen = VW_MAX_MESSAGE_LEN;

    if (vw_get_message(buf, &buflen)) // Se receber a mensagem analisa ela
    {
        if((buf[0] == '0') && (buf[1] == '1')&&(buf[2] == '0'))
        {
            digitalWrite(A0,1); //conforme a mensagem muda o estado da saída
            char *msg = "1001L#"; //envia mensagem para a central
            vw_send((uint8_t *)msg, strlen(msg));
            vw_wait_tx();
        }
        if((buf[0] == '0')&&(buf[1] == '1')&&(buf[2] == '1'))
        {
            digitalWrite(A0,0);
            char *msg = "0001L#";
            vw_send((uint8_t *)msg, strlen(msg));
            vw_wait_tx();
        }
        if((buf[0] == '0') && (buf[1] == '1')&&(buf[2] == '2'))
        {
            digitalWrite(A1,1);
            char *msg = "0101L#";
            vw_send((uint8_t *)msg, strlen(msg));
            vw_wait_tx();
        }
    }
}
```

```

    }
    if((buf[0] == '0') && (buf[1] == '1') && (buf[2] == '3'))
    {
        digitalWrite(A1,0);
        char *msg = "0001L#";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    if((buf[0] == '0') && (buf[1] == '1') && (buf[2] == '4'))
    {
        digitalWrite(A2,1);
        char *msg = "0011L#";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    if((buf[0] == '0') && (buf[1] == '1') && (buf[2] == '5'))
    {
        digitalWrite(A2,0);
        char *msg = "0001L#";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    delay(50);
}

char c = Serial.read(); //leitura do bluetooth
Serial.println(c);

if(c == 'A') //se receber via bluetooth o caractere
{
    digitalWrite(A0,1); //muda a saída
    char *msg = "1001L#"; //envia para a central essa mudança
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'a')
{
    digitalWrite(A0,0);
    char *msg = "0001L#";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'B')
{
    digitalWrite(A1,1);
    char *msg = "0101L#";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}

```



```

if(c == 'b')
{
    digitalWrite(A1,0);
    char *msg = "0001L#";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'C')
{
    digitalWrite(A2,1);
    char *msg = "0011L#";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'c')
{
    digitalWrite(A2,0);
    char *msg = "0001L#";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'D')
{
    char *msg = "020";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'd')
{
    char *msg = "021";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'E')
{
    char *msg = "022";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'e')
{
    char *msg = "023";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}
if(c == 'F')
{
    char *msg = "024";
    vw_send((uint8_t *)msg, strlen(msg));
    vw_wait_tx();
}

```

```

    }
    if(c == 'f')
    {
        char *msg = "025";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    if(c == 'G')
    {
        char *msg = "030";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    if(c == 'g')
    {
        char *msg = "031";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    if(c == 'H')
    {
        char *msg = "032";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    if(c == 'h')
    {
        char *msg = "033";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    if(c == 'I')
    {
        char *msg = "034";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    if(c == 'i')
    {
        char *msg = "035";
        vw_send((uint8_t *)msg, strlen(msg));
        vw_wait_tx();
    }
    }

    delay(50);
}

```

APÊNDICE E – PROGRAMAÇÃO MÓDULO 2

```
#include <VirtualWire.h>
```

```
#include <IRremote.h> //biblioteca do IR
```

```
int RECV_PIN = 7; // Declarando o pino do controle remoto
```

```
IRrecv irrecv(RECV_PIN);
```

```
decode_results results;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  irrecv.enableIRIn(); // iniciando o receptor de infravermelho
```

```
  Serial.println("setup");
```

```
  pinMode (A0, OUTPUT); //define os pinos de saída
```

```
  pinMode (A1, OUTPUT);
```

```
  pinMode (A2, OUTPUT);
```

```
  vw_setup(2000); // Bits per sec
```

```
  vw_set_rx_pin (8); // define o pino receptor para RF433 e RS485
```

```
  vw_set_tx_pin (9); //define o pino transmissor para RF433 e RS485
```

```
  vw_rx_start();
```

```
}
```

```
void loop()
```

```
{
```

```
  if (irrecv.decode(&results)) // se receber dados pelo IR
```

```
  {
```

```
    Serial.println(results.value, DEC);
```

```
    if (results.value == 1 || results.value == 2049) //executa acionamento conforme  
    dado recebido
```

```
    {
```

```
      digitalWrite(A0,HIGH);
```

```
    }
```

```
    if (results.value == 2 || results.value == 2050)
```

```
    {
```

```
      digitalWrite(A0,LOW);
```

```
    }
```

```
    if (results.value == 3 || results.value == 2051)
```

```
    {
```

```
      digitalWrite(A1,HIGH);
```

```
    }
```

```

if (results.value == 4 || results.value == 2052)
{
    digitalWrite(A1,LOW);
}
if (results.value == 5 || results.value == 2053)
{
    digitalWrite(A2,HIGH);
}
if (results.value == 6 || results.value == 2054)
{
    digitalWrite(A2,LOW);
}
irrecv.resume();
}

```

```

uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = VW_MAX_MESSAGE_LEN;
if (vw_get_message(buf, &buflen)) //se receber mensagem da central ou dos
módulos via RF433 ou RS485

```

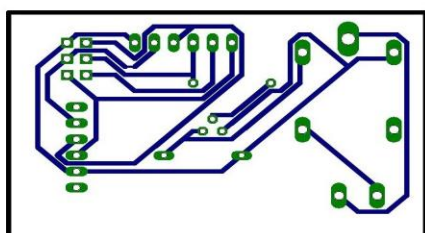
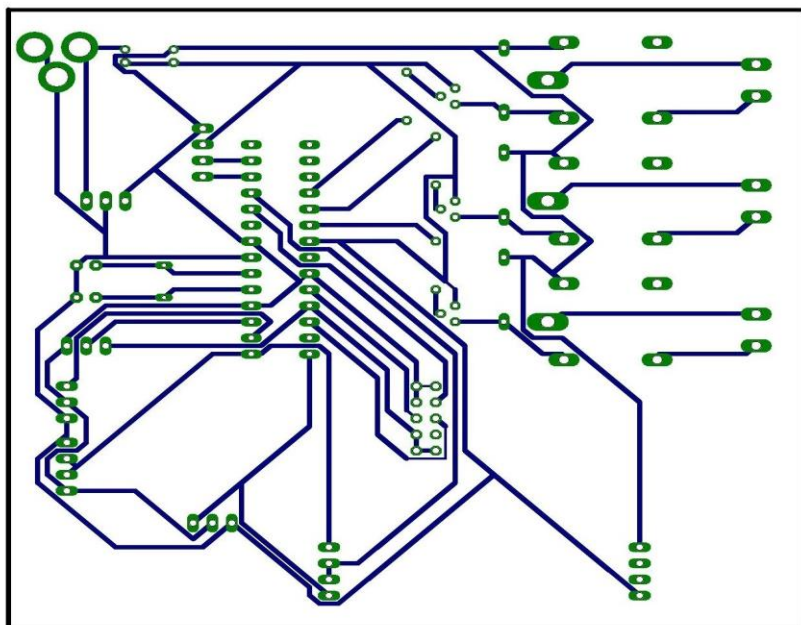
```

{
    int i;
    Serial.print("Got: ");
    for (i = 0; i < buflen; i++)
    {
        Serial.print(char(buf[i]));
    }
    Serial.println("");
    if((buf[0] == '0') && (buf[1] == '2')&&(buf[2] == '0'))
    {
        digitalWrite(A0,1);
    }
    if((buf[0] == '0')&&(buf[1] == '2')&&(buf[2] == '1'))
    {
        digitalWrite(A0,0);
    }
    if((buf[0] == '0') && (buf[1] == '2')&&(buf[2] == '2'))
    {
        digitalWrite(A1,1);
    }
    if((buf[0] == '0')&&(buf[1] == '2')&&(buf[2] == '3'))
    {
        digitalWrite(A1,0);
    }
}

```

```
    }  
    if((buf[0] == '0') && (buf[1] == '2')&&(buf[2] == '4'))  
    {  
        digitalWrite(A2,1);  
    }  
    if((buf[0] == '0')&&(buf[1] == '2')&&(buf[2] == '5'))  
    {  
        digitalWrite(A2,0);  
    }  
    delay(50);  
}  
}
```


APÊNDICE G – PCB MÓDULOS



APÊNDICE H – APLICATIVO ANDROID

Para realização do aplicativo a ser usado em um dispositivo com sistema operacional android foi utilizado o *software* MIT App Inventor 2. Este *software* é encontrado no site: <http://appinventor.mit.edu/explore/>. Não é necessário baixá-lo para o computador, ele é feito inteiramente no site. Ao iniciar um projeto é necessário colocar todos os elementos que se deseja utilizar na tela, por exemplo os botões que irão acionar as lâmpadas, as imagens, e os *labels*, que seriam campos de texto.

Na Figura 33 é possível visualizar o ambiente de trabalho e a tela com os elementos citados anteriormente.

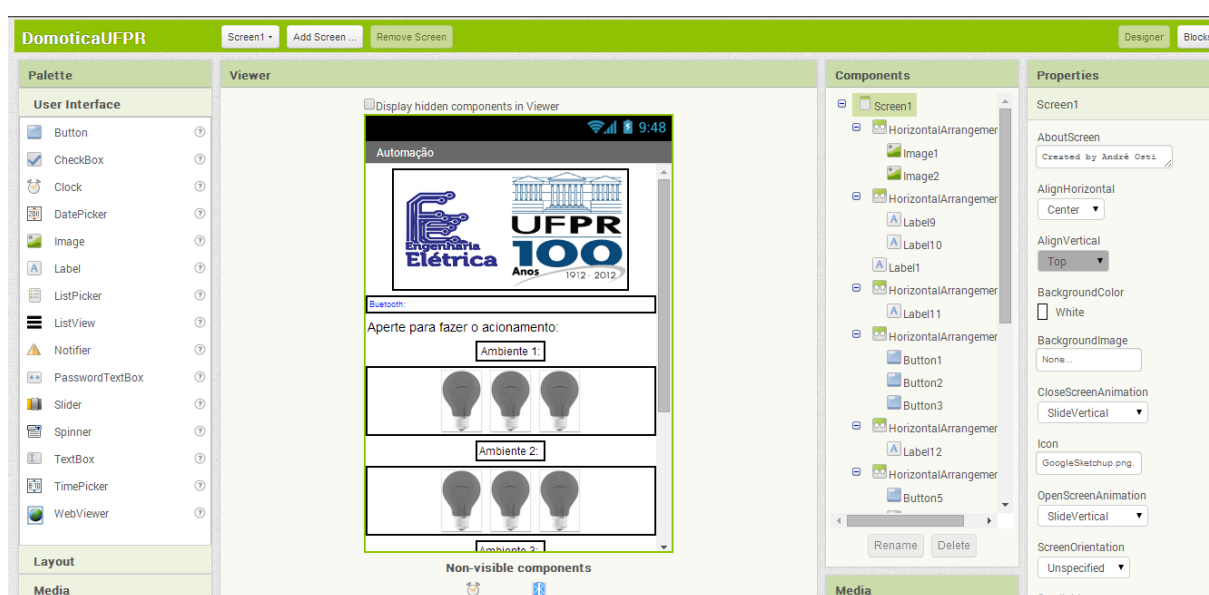


FIGURA 33 – AMBIENTE DE TRABALHO PARA O APLICATIVO
FONTE: O AUTOR (2014).

A partir do momento que se monta os elementos que se deseja ter na tela é necessário programar o que cada elemento irá realizar. Apertando no botão superior direito “Blocks” é possível mudar para a página de programação. A programação é feita por linguagem de blocos com as funções da linguagem de programação em C.

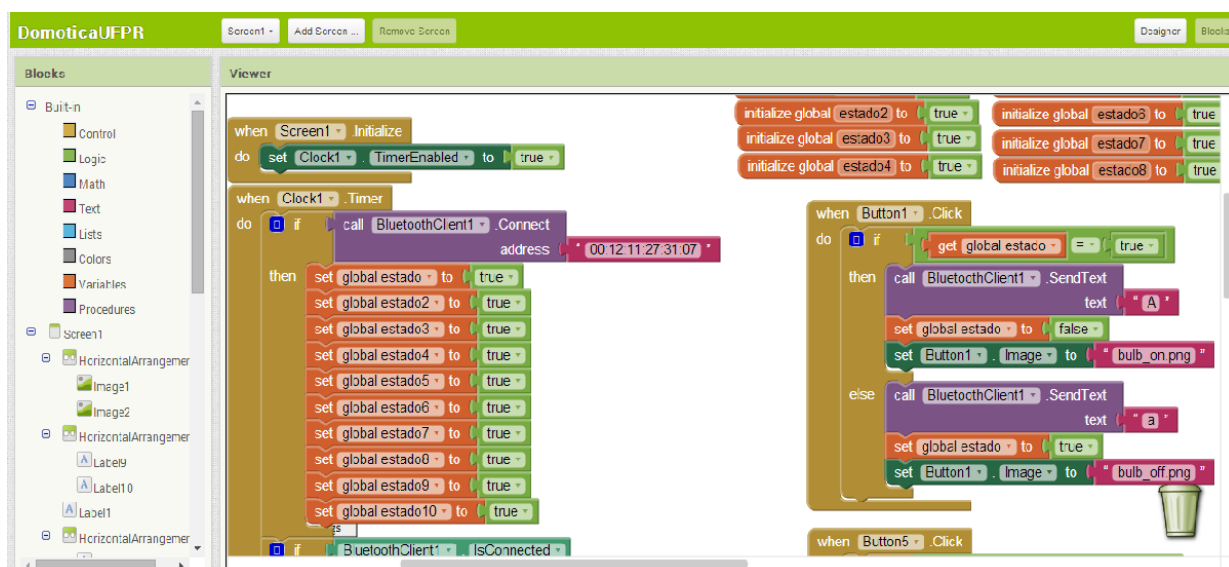


FIGURA 34 – PROGRAMAÇÃO PARA O APLICATIVO
FONTE: O AUTOR (2014).

Para realizar a conexão com o dispositivo bluetooth do módulo do ambiente é necessário saber qual o endereço deste dispositivo, este endereço é obtido ao parear o dispositivo com o celular, no caso a seguir “00:12:11:27:31:07”, com isso se programa para realizar a conexão conforme Figura 35.

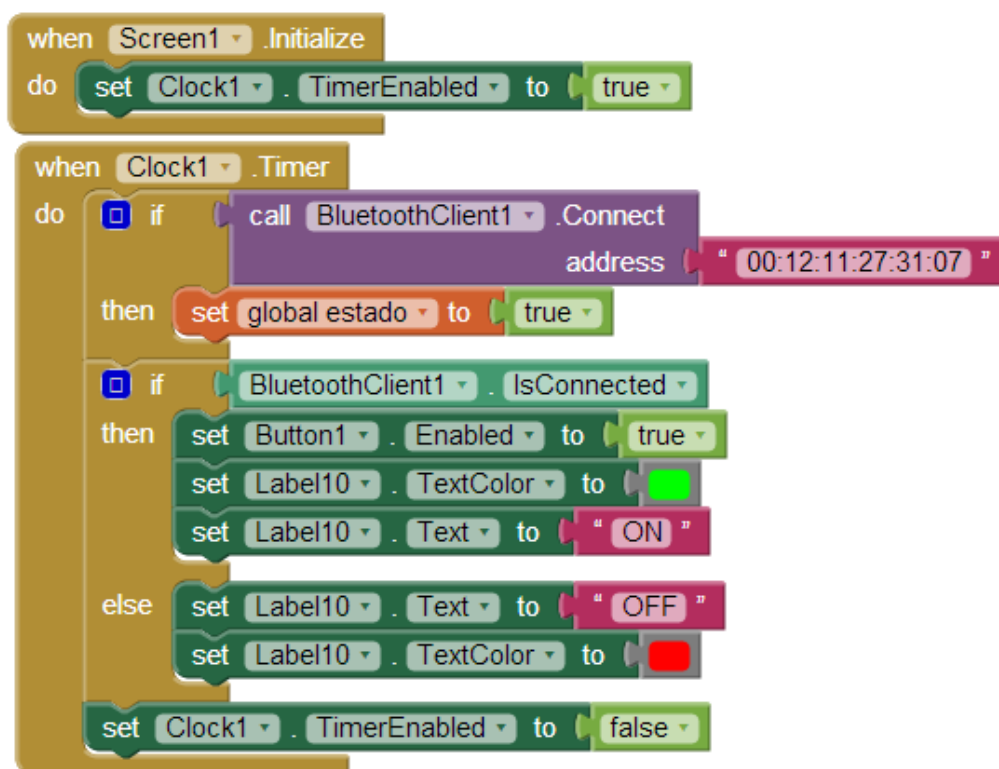


FIGURA 35 – PROGRAMAÇÃO CONEXÃO BLUETOOTH
FONTE: O AUTOR (2014).

O objetivo deste aplicativo é enviar comandos de acendimento e desligamento das lâmpadas do sistema. Ao pressionar um botão é enviado via *bluetooth* o comando para acender “A” e ao pressionar novamente é enviado o comando para desligar “a”. O microcontrolador já está programado para receber esses comandos.

É necessário criar uma variável de estado do botão (global estado = true), essa variável é responsável por mudar alternadamente os comandos de acionamento ao pressionar o botão.

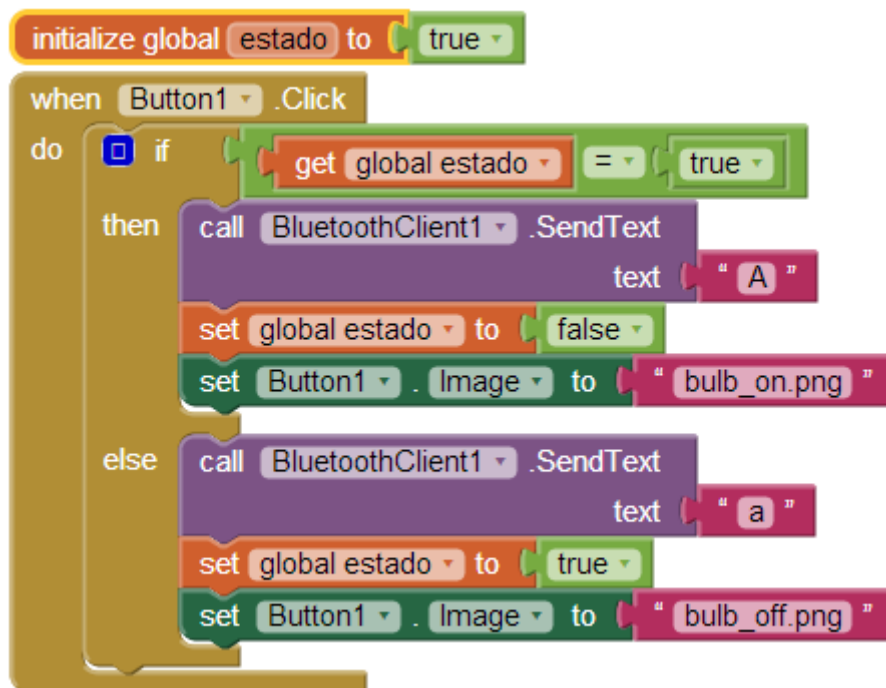


FIGURA 36 – PROGRAMAÇÃO ACIONAMENTO POR BOTÕES
FONTE: O AUTOR (2014).

Ao final da programação do aplicativo é necessário criar o pacote para instalação no celular ou tablet. Para isso é necessário acessar o menu Build e depois apertar em App (save .apk to my computer).

ANEXO A – DHT22

Power supply: 3.3-5.5V DC;
Output signal digital signal via 1-wire bus;
Sensing element Polymer humidity capacitor;
Operating range humidity 0-100%RH; temperature -40~80Celsius;
Accuracy humidity +-2%RH(Max +-5%RH); temperature +-0.5Celsius;
Resolution or sensitivity humidity 0.1%RH; temperature 0.1Celsius;
Repeatability humidity +-1%RH; temperature +-0.2Celsius;
Humidity hysteresis +-0.3%RH;
Long-term Stability +-0.5%RH/year;
Interchangeability fully interchangeable;

ANEXO B – MAX485

O chip MAX485 é um transceptor de baixa potência e taxa de variação limitada usado para comunicação RS-485. Ele trabalha em uma única fonte de alimentação de +5 V e a corrente nominal é de 300 mA. Adotando comunicação half-duplex para implementar a função de conversão de nível TTL em nível RS-485, pode atingir uma taxa de transmissão máxima de 2.5Mbps. O MAX485 transceptor se desempenha entre 120 μ A e 500 μ A o fornecimento nas condições descarregados ou totalmente carregado quando o driver está desativado. O condutor é limitado para a corrente de curto-circuito e as saídas de driver pode ser colocado em um estado de alta impedância através do circuito de desligamento térmico. A entrada do receptor tem um recurso à prova de falhas que garante a lógica de saída alta, se a entrada for de circuito aberto. Além disso, ele tem um bom desempenho anti-interferências (ITEADSTUDIO,2010).

ANEXO C – nrf24l01

Características do nRF24L01

a) Radio

- Worldwide 2.4GHz ISM band operation
- 126 RF channels
- Common RX and TX pins
- GFSK modulation
- 1 and 2Mbps air data rate
- 1MHz non-overlapping channel spacing at 1Mbps
- 2MHz non-overlapping channel spacing at 2Mbps

b) Transmitter

- Programmable output power: 0, -6, -12 or -18dBm
- 11.3mA at 0dBm output power

c) Receiver

- Integrated channel filters
- 12.3mA at 2Mbps
- -82dBm sensitivity at 2Mbps
- -85dBm sensitivity at 1Mbps
- Programmable LNA gain

d) RF Synthesizer

- Fully integrated synthesizer
- No external loop filter, VCO varactor diode or resonator
- Accepts low cost ± 60 ppm 16MHz crystal

e) Enhanced ShockBurst

- 1 to 32 bytes dynamic payload length
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe MultiCeiver™ for 1:6 star networks

f) Power Management

- Integrated voltage regulator
- 1.9 to 3.6V supply range
- Idle modes with fast start-up times for advanced power management
- 22uA Standby-I mode, 900nA power down mode
- Max 1.5ms start-up from power down mode
- Max 130us start-up from standby-I mod