UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

SISTEMA DE AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO COM USO DE RÁDIO FREQUÊNCIA

TRABALHO DE CONCLUSÃO DO CURSO DE CONTROLE E AUTOMAÇÃO

Gilberto Schneider

Santa Maria, RS, Brasil

2014

SISTEMA DE AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO COM USO DE RÁDIO FREQUÊNCIA

Gilberto Schneider

Trabalho de Conclusão de Curso apresentado ao curso de controle e automação, como parte necessária para a obtenção de grau do Centro de Tecnologia, da Universidade Federal de Santa Maria (UFSM,RS).

Orientador: Prof. Dr. Eng. Frederico Menine Schaf

Santa Maria, RS, Brasil

2014

Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Engenharia de Controle e Automação

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Conclusão de Curso

SISTEMA DE AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO COM USO DE RÁDIO FREQUÊNCIA

elaborado por
Gilberto Schneider

como parte necessária para a
Obtenção de Grau

COMISSÃO EXAMINADORA:

Frederico Menine Schaf
(Presidente/Orientador)

Carlos Henrique Barriquello

Claiton Moro Franchi

Santa Maria, 15 de dezembro de 2014.

RESUMO

Este trabalho apresenta uma proposta de um sistema de automação residencial, que utilizará como interface homem-máquina um dispositivo que possua o sistema operacional Android. Este celular/*tablet* se comunica com uma central instalada na residência, via Internet. A central por sua vez faz a comunicação com um microcontrolador central via *Bluetooth*. A partir do microcontrolador central a mensagem é distribuída por uma rede de comunicação sem fio que conecta diversos controladores.

Os controladores são dispositivos dotados de microcontroladores e com comunicação sem fio que executam as tarefas enviadas via interface homem-máquina. Para a finalidade de demonstrar a topologia do trabalho alguns dispositivos possuem LED's infravermelhos para controlar equipamentos com controle remoto convencional e outros possuem relés para acionamento de cargas diversas.

Com estes controladores, o sistema é capaz de comandar dispositivos acionados por controle remoto do tipo infravermelho, como por exemplo, televisões, aparelhos de DVD, condicionadores de ar entre outros. Também é possível controlar a iluminação da casa, ligando e desligando as lâmpadas que compõem esse sistema. Através do controle dos sistemas de iluminação, condicionadores de ar e multimídia de cada ambiente, o aplicativo é capaz de gerar modos de cena.

A rede de comunicação sem fio utilizada possui uma arquitetura tipo arvore fazendo o roteamento da mensagem desde a central até o ponto desejado, através dos outros controladores. Toda a comunicação sem fio recebeu criptografia baseada em uma senha gerada por data afim de proteger o pacote de dados assim garantido que o usuário fique a salvo de possíveis invasões intencionais ou não ao seu sistema.

O principal aspecto do projeto é a modularidade dos seus controladores, pois no momento em que a rede é implementada pode haver outros dispositivos desse tipo com diferentes funções para controlar diferentes aparelhos do nosso ambiente.

O sistema se mostrou economicamente viável e de simples instalação já que não é necessário cabeamento entre os controladores e a central. O algoritmo de criptografia na comunicação conferiu segurança para as mensagens enviadas via rádio frequência e o roteamento das mensagens garantiu o aumento no alcance da rede.

Palavras Chaves: Automação Residencial, Android, Arduino, Criptografia.

ABSTRACT

This work presents a proposal of a home automation system, which will use as human-machine (HMI) interface a mobile device with Android operating system. This mobile/tablet communicates with a central (controller) installed in the residence via the Internet. The central communicates with a central microcontroller via Bluetooth. From the central microcontroller the message is distributed over wireless communication network that connects multiple controllers.

The controllers are microcontrollers with wireless communication performing tasks via HMI. For the purpose of demonstrating the working topology some devices (modules) are equipped with infrared LED's to control home electronics with a conventional remote control. Other modules are equipped with relays to drive different electrical loads.

With these controllers, the system is capable of operate devices driven by infrared remote control type, such as televisions, DVD players, air conditioners and others. It is also possible to the lighting of the house, turning on and off lights. Through the control of lighting systems, air conditioners and multimedia, in each room, the application is able to generate scene modes.

The employed wireless communication network uses the tree (hierarchical) architecture type. Routing the message from the central to end devices through point-to-point communications. All wireless communications received is encrypted by an algorithm based on password generated by date in order to protect the data packet providing safety against possible intentional invasions or not your system.

The main aspect of the project is the modularity of its drivers (modules), because at the time the network is implemented there could be other devices with different functions to control different devices of our system.

The system was economically viable and has simple installation as it is not necessary cabling between the controllers and the central. The encryption algorithm in communication has given security to messages sent via radio frequency and routing of messages secured the increase in range of the network.

Key Words: Home Automation, Android, Arduino, Encryption.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Os processos de automação consistem na redução do esforço, e da mão de obra para a realização de uma tarefa. Comumente o termo de automação é vinculado à área industrial, porém com o desenvolvimento de novas tecnologias que tornam seu uso economicamente mais viável passou-se a existir uma linha para residências.

A automação residencial como abordada neste projeto se difere de pequenas melhorias feitas em eletrodomésticos isolados para tornar a operação destes, automática. Por exemplo, uma máquina de lavar ou um forno com temporizador. Neste trabalho a automação residencial será vista do ponto de vista da integração de dispositivos a uma rede de forma a tornar o ambiente mais confortável. Essa área de automação residencial, também conhecida como domótica, se baseia em integrar os diversos equipamentos utilizados em casa, de forma que esses possam ser comandados por um único dispositivo, ou por uma rede destes, de forma simples e inteligente (TEZA, 2002).

Um grande problema ainda enfrentado pela automação residencial é seu alto custo de instalação e por vezes a baixa flexibilidade, onde os dispositivos de automação são feitos para serem aplicados com equipamentos do mesmo fabricante. O objetivo deste trabalho é propor a automação, que já é presente em âmbitos industriais e comerciais devido ao seu evidente aumento de produtividade e redução de custos, para dentro das residências de forma simples e com baixo custo.

Para que fosse possível alcançar o objetivo do trabalho foi proposta uma mudança para a implementação da rede de controladores da automação residencial. Como o modelo de automação praticado atualmente, tanto residencial como industrial, opera com um processamento central conectado a atuadores sem poder computacional é necessário o recebimento de comandos via uma rede de comunicação cabeada. Este trabalho utiliza uma rede sem fio com controladores microcontrolados, eliminando os custos de cabeamento e reduzindo os de instalação.

O uso de redes sem fio passou a ser viável no últimos anos devido a exponencial redução de custos deste tipo de tecnologia e a maturidade obtida com os anos de pesquisa. O aumento de atratividade por de redes do tipo *WLAN* (Rede de Comunicação Local Sem Fio, acrônimo em inglês) pode ser visualizado na Figura 1 retirada do artigo de (LIAO, WELLET e EMMEL, 2005) que também mostra a redução de custos.

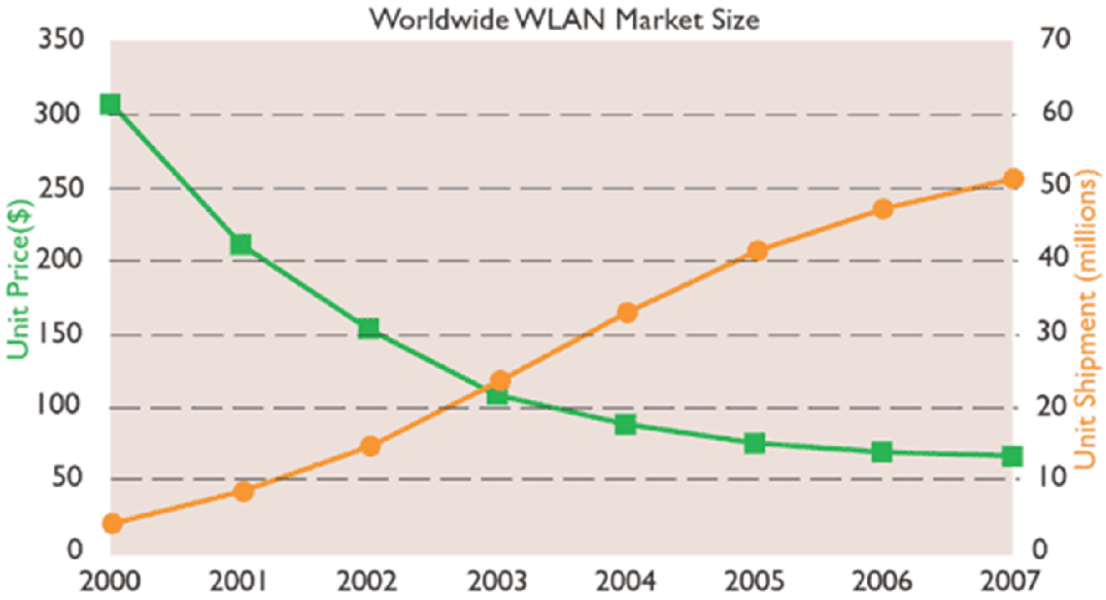

Figura 1 - Custo e utilização de redes sem fio
Fonte: (LIAO, WELLET e EMMEL, 2005)

O artigo também ressalta alguma preocupações que são consideradas no uso de uma rede sem fio. As principais ressalvas feitas tanto em (LIAO, WELLET e EMMEL, 2005) quanto em (GUNGOR e HANCKE, 2009), estão relacionadas a problemas intrínsecos do uso de um meio físico aberto. Onde são listadas:

– Garantia de entrega de mensagem com tempo de atraso conhecido;

– Determinismo Temporal;

– Segurança da Rede;

– Suporte a um grande número de dispositivos.

Como o trabalho proposto está focado em automação residencial e predial, parte-se do pressuposto que não é necessário um determinismo temporal das mensagens já que o sistema não deve ser utilizado para controle de processos críticos. A variação no tempo de atraso das mensagens não implica em danos ao usuário nem à aplicação, pois é praticamente indiferente se um comando para iluminação for entregue alguns milissegundos antes ou depois do previsto.

O tópico que diz respeito ao número de dispositivos na rede é importante, já que pode-se estar falando da automação de um ambiente com poucos dispositivos ou da automação de um prédio ou casa de grande porte. Isso implica diretamente na rede e nos módulos de comunicação escolhidos para o trabalho, pois estes devem ser capazes de se adaptar a ambos os casos mantendo a viabilidade do projeto.

Outro ponto relevante para a comunicação escolhida no trabalho é a segurança e integridade da comunicação devido a seu uso para controlar os mais diversos atuadores dentro de uma residência. Como já citado anteriormente o meio físico pelo o qual o sinal viaja, o ar, é

livre. Assim os dados que por ali trafegam estão sujeitos a interferência externa intencional e não intencional. Para interferências do tipo não intencional apenas uma verificação de integridade da mensagem se faz suficiente, porém para as do tipo intencional é necessária a adição de um mecanismo mais complexo. O trabalho apresenta um método de criptografia com senha única baseada no tempo e encriptação padrão *DES* (Encriptação Padrão de Dados, acrônimo em inglês).

Para a interface do usuário com os controladores foi desenvolvido um aplicativo para Android[1]. A escolha deste sistema operacional deve-se por sua grande parcela do mercado de dispositivos móveis, por possuir código aberto e plataformas de desenvolvimento para aplicativos serem gratuitas. O *hardware* utilizado para o sistema de supervisão do projeto consiste então em um aparelho com Android, que serve de interface homem-máquina rodando o aplicativo desenvolvido ao longo do projeto. Com o aplicativo é possível que seja feito o acesso remoto do sistema de automação residencial, via Internet.

Os dispositivos utilizados para atuar no ambiente são Arduinos[2] que tem o papel de receber os comandos vindos de um dispositivo com Android e executar as ações correspondentes a esses comandos. O projeto foi idealizado para o controle de mais de um ambiente, desta forma cada ambiente pode possuir diversos controladores, e estes estarem distantes do dispositivo com Android. Para não violar a diretiva de baixo custo que norteia o projeto é necessário que os módulos que fazem a comunicação sem fio sejam de baixa potência, logo é preciso que seja feito o roteamento das mensagens. Redes roteadas reduzem o custo total da infraestrutura de comunicação se comparada a uma que tem conexões apenas ponto a ponto como é mostrado pela lei de Metcalfe (METCALFE, 2013).

Os diversos controladores são capazes de executar tarefas como controle de: iluminação, condicionamento de ar e eletrodomésticos que já possuem algum tipo de controle remoto. Porém em longo prazo, devida a modularidade da topologia, podem vir a ser adicionados outros controladores que façam tarefas diversas, automatizando uma infinidade de dispositivos residenciais.

Nas seguintes seções desse trabalho serão explicados os processos de desenvolvimento do aplicativo, e dos *hardwares* utilizados no projeto, mostrando os desafios e tecnologias envolvidas. Para a validação dos algoritmos e da topologia proposta no projeto foram desenvolvidos protótipos, que foram submetidos a testes relacionados à criptografia das mensagens RF e sua interpretação, teste de alcance dos módulos, ensaio de longa duração e estimativa de custos para comprovação da viabilidade econômica.

[1] - Site oficial do sistema operacional Android: www.android.com/

[2] - Site oficial do projeto Arduino: www.arduino.cc/

O trabalho está organizado da seguinte forma: na seção 2 está uma revisão teórica sobre automação residencial e as tecnologias utilizadas atualmente; na seção 3 está descrito o desenvolvimento do projeto e das tecnologias propostas; a seção 4 é composta por uma apresentação de protótipos e resultados experimentais que incluem análise de custo; na seção 5 estão as conclusões retiradas desse trabalho; por fim existe uma seção com apêndices descrevem melhor alguns conceitos e tecnologias utilizadas durante o desenvolvimento do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão explicados os principais conceitos utilizados ao longo do trabalho. Serão mostrados também alguns dos sistemas utilizados atualmente na automação residencial para que nas próximas seções possa ser feito um paralelo entre a tecnologia atual e a proposta pelo trabalho.

Este capitulo está divido entre duas partes principais: Inicialmente um estado da arte da automação residencial contendo uma explicação sobre as tecnologias mais utilizadas atualmente para automação residencial a nível comercial, suas peculiaridades e alguns custos. A outra parte versa sobre redes de comunicação com uma introdução ao modelo de camadas ISO/OSI onde será mostrado como classificar cada parte de uma rede de comunicação e uma análise do uso de criptografia em redes de comunicação.

2.1 Automação Residencial na Atualidade

Hoje em dia vê-se cada vez mais dispositivos eletrônicos apresentar algum tipo de comunicação que os torna integrados a uma interface com o usuário. Principalmente após o advento e popularização de *smartphones* é possível notar um crescimento cada vez mais acelerado em tecnologias de automação e controle remoto para objetos do cotidiano.

O crescimento do uso de celulares como interfaces para a automação de equipamentos se deu como um processo natural, devido ao fato de ser um dispositivo cotidiano que nos acompanha em praticamente todos os momentos. Entre *smartphones*, os que apresentaram maior liberdade para a criação de novos aplicativos e consequentemente um maior volume de aplicativos voltados a automação foram os que possuíam a plataforma Android como sistema operacional.

Essa maior liberdade se deve as ferramentas de desenvolvimento serem gratuitas e o sistema operacional apresentar código aberto, dando possibilidade a desenvolvedores amadores tentarem novas aplicações e garantindo mais flexibilidade a programadores avançados.

Nas próximas subseções serão mostradas três vertentes de tecnologias utilizadas para automação residencial: Automação de dispositivos discretos; Automação com dispositivos cabeados e Automação com dispositivos integrados via rede de comunicação sem fio.

2.1.1 Automação de Dispositivos Discretos

Automação de dispositivos discretos é o tipo de automação residencial mais comum que se observa hoje em dia. Sua principal característica é a comunicação entre uma interface homem-máquina e um dispositivo a fim de conferir novos recursos como a possibilidade de acionamento remoto, temporização, agendamento, definição de modos de operação, entre outros.

São exemplos deste tipo de automação tecnologias como a da Philips HUE[3] que é uma lâmpada "inteligente", ou um conjunto de lâmpadas "inteligentes". Essas lâmpadas (de LED) são dotadas de uma comunicação Zigbee[4] com uma central também vendida pela Philips, chamada de HUE HUB, e que se conecta então com a Internet possibilitando o controle das lâmpadas via rede Wi-fi ou Internet através de um celular com Android ou iOS. Porém, é possível notar que esta automação fica restrita a apenas um tipo de dispositivo, sendo o aplicativo desenvolvido para celular totalmente dedicado. Uma imagem dos dispositivos citados da Philips está na Figura 2.

Figura 2 - Hub e Lâmpadas HUE
Fonte: Site da Philips HUE[3]

Outro exemplo de dispositivos discretos que sofreram integração com algum tipo de rede são os multimídias. Atualmente é fácil encontrar sons e receptores vídeo para televisores que apresentem algum tipo de canal com o celular para reprodução direta de músicas e imagens sem a necessidade de conexão via cabo entre o celular e o equipamento multimídia. O Chromecast[5] é um exemplo deste tipo de dispositivo, ele permite o *streaming* de mídia utilizando um equipamento do tamanho de um *pendrive* que se conecta à porta *HDMI* do televisor.

[3] – Site oficial da HUE em português: http://www2.meethue.com/pt-pt/

[4] – Site oficial da aliança Zigbee: http://zigbee.org/

[5] – Site oficial do Chromecast: https://www.google.com.br/chrome/devices/chromecast/

Basta usar um *smartphone* Android, um *tablet*, um iPhone, um iPad, um notebook Windows ou Mac ou um Chromebook para transmitir os programas de entretenimento e seus aplicativos diretamente para a tela do televisor. A Figura 3 mostra a ideia de funcionamento do Chromecast.

Figura 3 - Demonstração do Chromecast
Fonte: Site do Chromecast[5]

Assim como a lâmpada HUE o Chromecast é também dedicado a automatizar apenas um dispositivo e não voltado para o conceito mais amplo de domótica onde se está interessado em automatizar um ambiente ou até mesmo uma casa inteira.

2.1.2 Automação com Dispositivos Cabeados

A automação fazendo uso de dispositivos sem processamento que apenas recebam um sinal de potência via uma rede de cabos ainda é o tipo mais utilizado industrialmente e foi até poucos anos atrás o único tipo de automação utilizada no ambiente residencial. Ou ainda, o uso de dispositivos micro processados, mas com cabos para levarem a comunicação até esses pontos.

Este tipo de automação é caracterizado pelo baixo custo dos dispositivos controladores e elevados custos de instalação ocasionados pela necessidade de cabeamento. Geralmente são feitos de forma conjunta com a construção do imóvel numa tentativa de reduzir os custos.

O exemplo mais avançado desta tecnologia é o uso da própria rede elétrica como meio físico para os dados. Onde todos os dados são transmitidos pelos cabos de energia. Como exemplo deste tipo de automação está o padrão chamado de X10[6].

Existem diversos módulos disponibilizados por esta empresa para as mais diversas aplicações, como *dimmer* de lâmpadas, controle de cargas e até mesmo uma interface USB com o computador. Alguns exemplos de controladores operando em X10 são mostrados na Figura 4. Onde a linha vermelha indica uma comunicação do tipo PLC(Comunicação por linha de energia)

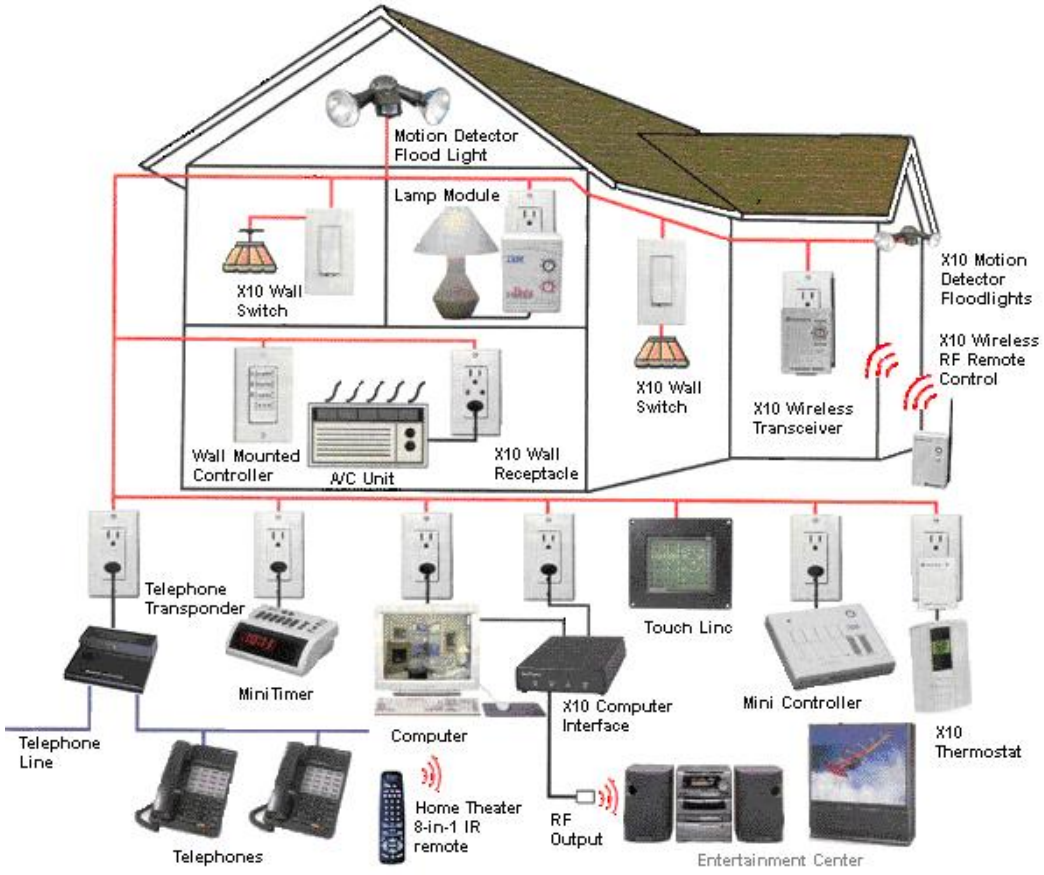

Figura 4 - Aplicações do X10
Fonte: Site do X10[6]

Se formos analisar o tipo mais tradicional, completamente cabeado sem rede de comunicação, existem grandes fabricantes como a General Eletric e a Legrand que disponibilizam painéis já com *softwares* para que seja feita uma automação residencial. O grande problema deste tipo de automação é a baixa flexibilidade e o alto custo com cabeamento como já referido anteriormente. O sistema de automação residencial mais divulgado da GE é o

[6] – Site oficial da X10 para automação residencial: http://www.x10.com/x10-home-automation.html

HabiTEQ[7] e o da Legrand é o on-Q[8]. Ambos disponibilizam diversos tipos de controladores e sensores para serem integrados a suas centrais. Porém, o cabeamento é especifico para cada subcentral de dispositivos e cada dispositivo atuador ou sensor necessita de um cabo de sinal/potência.

2.1.3 Automação com Dispositivos Sem Fio

A automação com dispositivos sem fio é constituída de dispositivos com microcontroladores conectados a redes de comunicação sem fio, por onde são recebidos os comandos provenientes do usuário. É sem dúvida a área que mais se desenvolve nos últimos anos e também é a área de estudo deste trabalho.

As principais características deste tipo de automação é o baixo custo de instalação e a possibilidade de controle distribuído, devido ao fato dos dispositivos possuírem poder de processamento.

Nos últimos anos o desenvolvimento deste tipo de automação foi acelerado com a criação de padrões de protocolo e dispositivos para comunicação sem fio que apresentam baixo custo e facilidade de programação. Dentre esses tipos de rede pode se ressaltar o já citado Zigbee e o Z-Wave[9].

Um exemplo de tecnologia ainda mais complexa é o WINK[10] que possui uma central capaz de se conectar com dispositivos *bluetooth*, Zigbee, Z-Wave entre outras. Um exemplo de automação residencial utilizando o WINK, é possuir uma lâmpada GE LINK, uma tomada do QUIRKY+GE e um HUB WINK. Controlando tudo através de um *smartphone*, com um único aplicativo. Na Figura 5 estão alguns exemplos de aplicação do WINK que são informados no site do fabricante.

[7] – Site do HabiTEQ: http://www.gepowercontrols.com/eu/product_portfolio/residential/building_automation/habiteq.html

[8] – Site do on-Q: http://www.legrand.us/onq/home-automation/

[9] – Site oficial do Z-Wave: http://www.z-wave.com/

[10] – Site do WINK: http://www.wink.com/

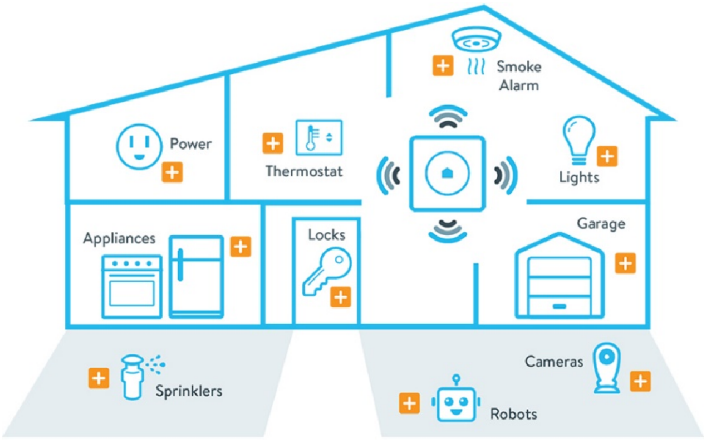

Figura 5 - Exemplos de aplicação com WINK
Fonte: WINK[10]

2.1.4 Custo de dispositivos de automação

Como o trabalho visa construir uma rede de dispositivos de baixo custo, é interessante uma análise sobre os custos da automação residencial na atualidade para fins de comparativo. Foram selecionados alguns dispositivos anteriormente citados, os quais o fabricante informa o custo individual de cada dispositivo. É importante ressaltar que geralmente a automação residencial possui um preço por pacote e não por dispositivo, por isso o trabalho dispõe de apenas alguns tipos de dispositivos de alguns fabricantes.

2.1.4.1 Custo de Dispositivos Discretos

Os dois tipos de dispositivos citados anteriormente nesta categoria tem o seu valor informado pelo fabricante conforme a Tabela 1.

Tabela 1 - Custo de Dispositivos Automatizados

Produto	**Custo(R$)**
Kit 3 Lâmpadas HUE + HUB (*Starter Pack*)	1299,00[11]
Chromecast	249,00[12]

[11] – Preço obtido em 25/11/2014 as 19:00 no site: http://store.apple.com/br/product/HA779BR/A/l%C3%A2mpada-philips-hue-connected-starter-pack

[12] – Preço obtido em 25/11/2014 as 19:05 no site: https://www.google.com.br/chrome/devices/chromecast/

2.1.4.2 Custo da Automação com Dispositivos Cabeados

Para este tipo de automação foram considerados apenas os dispositivos que dispensam uma custo extra na instalação para que fosse feito um comparativo justo. Sendo assim foram utilizados para comparativos os produtos que utilizam X10, que utilizam como rede de comunicação a própria rede elétrica (tecnologia PLC – *Power Line Carrier*). Com os dispositivos descritos na Tabela 2 é possível controlar via computador, um ponto de iluminação e uma tomada com duas saídas.

Tabela 2 - Custo de dispositivos X10

Produto	Custo(R$)[13]
Transceiver USB	76,80
Controlador de Iluminação	76,80
Controle de Carga	89,60

2.1.4.3 Custo da Automação com Dispositivos Sem Fio

Foram avaliados para esta categoria os produtos da GE que interagem com o sistema WINK os preços estão mostrados na tabela abaixo:

Tabela 3 - Custo de dispositivos X10

Produto	Custo(R$)[14]
HUB WINK	128,00
Controlador de Carga	128,00
Lâmpada GE LINK	41,00

Com estes dispositivos é possível controlar uma carga genérica e uma lâmpada especial da GE, tudo via um único aplicativo para celular. O diferencial desse sistema é que existe uma infinidade de dispositivos compatíveis com o WINK e seu aplicativo.

[13] – Preço obtido no site da X10 EUA e convertido para reais com a cotação do dia (25/11/2014): http://www.x10.com/x10-home-automation

[14] – Preço obtido no site da WINK EUA e convertido para reais com a cotação do dia (25/11/2014): http://www.wink.com/products/

2.2 O Modelo ISO/OSI

Para facilitar a compreensão das redes de comunicação desenvolvidas ao longo do trabalho esta seção irá explicar a parte que se refere ao modelo de camadas da norma *ISO/OSI*. Com este modelo é possível classificar cada rede de comunicação quanto a seu meio físico, como é feito o roteamento da mensagem, a codificação dos dados e outros fatores relevantes para o entendimento de um protocolo de comunicação.

O modelo *ISO/OSI* foi criado com o objetivo de facilitar a conexão entre diferentes sistemas. É regulamentado pela *ISO* (Organização Internacional de Padrões, acrônimo em inglês), em sua norma ISO/IEC 7498-1 (INTERNATIONAL STANDARD, 1994), chamada de Interconexão de Sistemas Abertos (*OSI*, acrônimo em inglês).

2.2.1 Arquitetura OSI

A descrição da arquitetura da organização do modelo OSI é feita ao longo do capítulo sete da norma ISO/IEC 7498-1 e será resumido ao longo desta subseção. A arquitetura foi descrita em sete camadas que serão explicadas a seguir, sendo estas:

1. Camada Física;
2. Camada de Enlace de Dados;
3. Camada de Rede;
4. Camada de Transporte;
5. Camada de Sessão;
6. Camada de Apresentação;
7. Camada de Aplicação.

O fluxograma da Figura 6 dá uma ideia das etapas seguidas por um dado transferido através de comunicações que respeitam o modelo OSI, desde de sua transmissão a partir de um aplicativo até o recebimento em outro aplicativo. Levando em consideração a manutenção do sentido da mensagem, para que essa mantenha-se útil para o receptor.

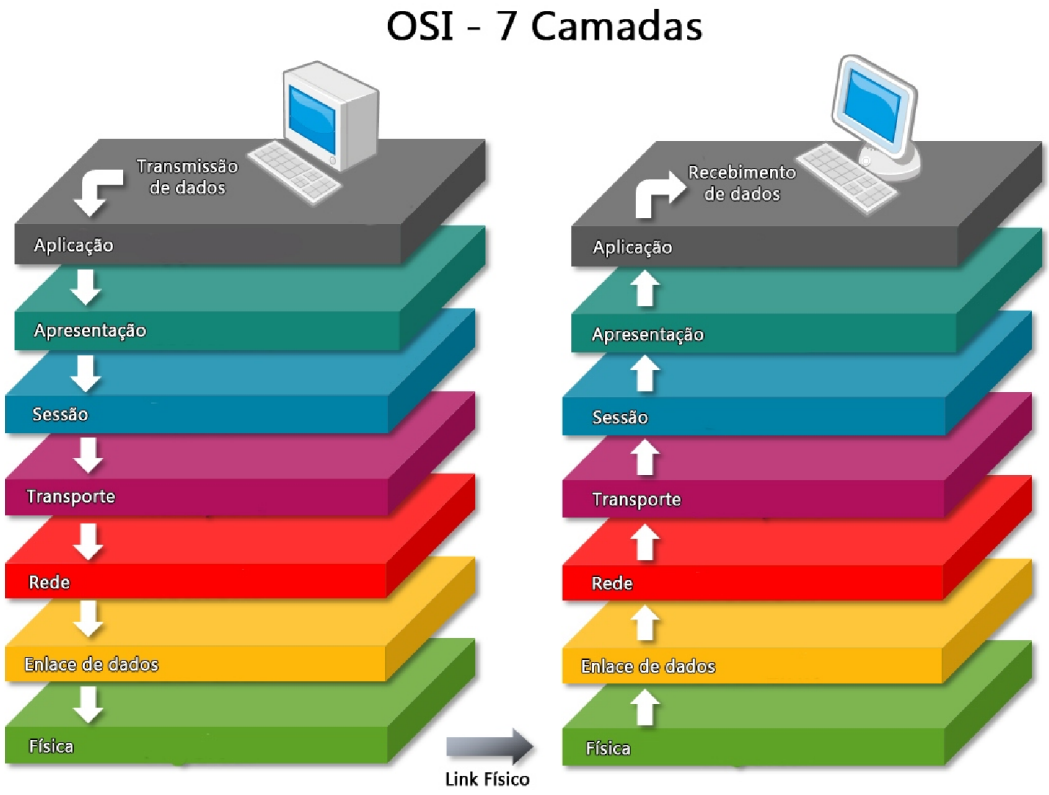

Figura 6 - Funcionamento do Modelo ISO/OSI

2.2.1.1 Camada Física

Está camada é o ponto mais baixo da arquitetura de qualquer comunicação e indica que tipo de meio físico será usado na transmissão dos dados. A camada física define as características técnicas dos dispositivos elétricos e ópticos (fios, conectores, tensões, taxa de dados *tranceivers,* entre outros) do sistema. Ela contém os equipamentos de cabeamento ou outros canais de comunicação que se comunicam diretamente com o controlador da interface de rede. Como características básicas é possível destacar então:

- Transfere bits através de um meio físico;
- Define as características elétricas e mecânicas do meio, taxa de transferência dos bits, tensões...;
- Limita a quantidade e velocidade de transmissão de informações na rede.

2.2.1.2 Camada de Enlace de Dados

Esta camada é responsável pela transmissão e recepção (delimitação) de quadros e pelo controle de fluxo. Ela também estabelece um protocolo de comunicação entre sistemas diretamente conectados. Em redes do padrão IEEE 802, esta camada é dividida em outras duas camadas: Controle de ligação lógica (LLC), que fornece uma interface para camada superior (rede), e Controle de acesso ao meio físico (MAC), que acessa diretamente o meio físico e controla a transmissão de dados.

2.2.1.3 Camada de Rede

A camada de Rede é responsável pelo endereçamento dos pacotes, convertendo endereços lógicos em endereços físicos, de forma que os pacotes consigam chegar corretamente ao destino. Essa camada também determina a rota que os pacotes irão seguir para atingir o destino, baseada em fatores como condições de tráfego da rede e prioridades. Esta camada está presente quando a rede possui mais de um segmento e, com isso, há mais de um caminho possível para um pacote de dados percorrer, sendo necessário o roteamento para que este chegue até o destino.

2.2.1.4 Camada de Transporte

A camada de transporte é responsável por usar os dados enviados pela camada de Sessão e dividi-los em pacotes que serão transmitidos para a Camada de Rede. No receptor, a camada de Transporte é responsável por pegar os pacotes recebidos da camada de Rede, remontar o dado original e assim enviá-lo à camada de Sessão.

Isso inclui controle de fluxo, ordenação dos pacotes e a correção de erros, tipicamente enviando para o transmissor uma informação de recebimento, informando que o pacote foi recebido com sucesso.

A camada de transporte separa as camadas de nível de aplicação (camadas 5 a 7) das camadas de nível físico (camadas de 1 a 3). A camada 4, Transporte, faz a ligação entre esses dois grupos e determina a classe de serviço necessária como: orientada a conexão e com controle de erro e serviço de confirmação; ou sem conexões e nem confiabilidade.

O objetivo final da camada de transporte é proporcionar serviço eficiente, confiável e de baixo custo. O *hardware* e/ou *software* dentro da camada de transporte e que faz o serviço é denominado entidade de transporte.

2.2.1.5 Camada de Sessão

A camada de sessão permite que duas aplicações em computadores diferentes estabeleçam uma sessão de comunicação. Nesta sessão, essas aplicações definem como será feita a transmissão de dados e coloca marcações nos dados que serão transmitidos. Se porventura a rede falhar, os computadores reiniciam a transmissão dos dados a partir da última marcação recebida pelo computador receptor.

2.2.1.6 Camada de Apresentação

A camada de Apresentação, também chamada camada de Tradução, converte o formato do dado recebido pela camada de Aplicação em um formato comum a ser usado na transmissão desse dado, ou seja, um formato entendido pelo protocolo usado. Um exemplo comum é a conversão do padrão de caracteres quando o dispositivo transmissor usa um padrão diferente do ASCII.

Para aumentar a segurança, pode-se usar algum esquema de criptografia neste nível, sendo que os dados só serão decodificados na camada 6 do dispositivo receptor.

2.2.1.7 Camada de Aplicação

A camada de aplicação é responsável por identificar e estabelecer a aplicação (programa) o qual será utilizado entre a máquina destinatária e o usuário como também disponibiliza os recursos (protocolo) para que tal comunicação aconteça. Por exemplo, ao solicitar a recepção de e-mail através do aplicativo de e-mail, este entrará em contato com a camada de Aplicação do protocolo de rede efetuando tal solicitação (POP3, IMAP). Tudo nesta camada é direcionado aos aplicativos.

2.3 A camada de Apresentação como garantia de segurança

Durante a criação do modelo OSI foi pensada uma camada para que os dados que fossem enviados entre diferentes sistemas conseguissem ser interpretados da forma correta. Com o passar dos anos os programadores perceberam que a codificação diferente das mensagens poderia na verdade não ser apenas um problema para que dois sistemas conseguissem manter uma comunicação.

Na verdade este tipo de técnica poderia ser aplicada propositalmente para evitar que dispositivos de terceiros pudessem compreender a mensagem. Esse tipo de técnica de embaralhar de forma controlada as mensagens é conhecido como criptografia ou encriptação.

Como o trabalho utilizará redes de comunicação sem fio, ou seja, um meio físico completamente suscetível a ataques, esta subseção irá abordar um pouco sobre criptografia e sua aplicação atualmente em redes residenciais.

2.3.1 História da Criptografia

A criptografia é uma técnica extremamente antiga, acredita-se que no Egito antigo algumas mensagens já possuíam tais técnicas. Praticamente todas as culturas até o dia de hoje se interessam em manter alguns tipos de informação sobre sigilo.

Quando pensamos em criptografia moderna, o primeiro exemplo é a famosa máquina Enigma criada pelo alemães e posteriormente modificada para manter mensagens militares seguras durante a segunda guerra mundial. Ela foi a primeira máquina automatizada para esta finalidade.

Após este tipo de técnica ser utilizada em períodos de guerra houve então o primeiro grande salto nesta área. Governos passaram a investir em pesquisas de novas técnicas e com o advento da computação, foram criados algoritmos de criptografia. O primeiro algoritmo de criptografia aberta a ser inventado foi o DES. (FIPS, 1999)

Inicialmente a criptografia era utilizada apenas para mensagens ultra secretas de órgãos militares, porém com o avanço do uso de computadores e redes de comunicação para finalidades que requeriam segurança, como operações bancárias e troca de mensagens industriais, diversas companhias passaram a adotar mecanismos de criptografia.

O próximo salto da criptografia ocorre com o uso de redes sem fio, que transmitem dados pessoais e até mesmo controlam funções importantes para o nosso dia a dia e podem facilmente ser comprometidas.

2.3.2 Uso da criptografia na automação

Atualmente apenas três das redes mais utilizadas para automação residencial possuem algum tipo de segurança ligada a suas mensagens, o *bluetooth*, o Zigbee e o recente *Z-Wave.* A capacidade de garantir uma mensagem inviolada para o Z-wave e Zigbee foram avaliadas pelo trabalho (KNIGHT, 2006), ambos os sistemas utilizam uma criptografia do tipo AES (Padrão de Criptografia Avançado, acrônimo em inglês) que é uma evolução do DES. É fácil perceber a carência de redes seguras na automação residencial, isto torna aplicações relacionadas à segurança residencial, como o controle de acesso e dispositivos de alarme que são colocados em rede, dispositivos praticamente inúteis ou até mesmo facilitadores para pessoas mal intencionadas.

As redes via cabo não representam grande preocupação desde que sejam do tipo ilhadas, ou seja, não acessíveis pelo mundo externo. Porém redes sem fio podem ser acessadas sem a necessidade de contato elétrico, logo estas devem ser dotadas de maior nível de segurança.

O *bluetooth* fornece tal segurança a conexão, mas ainda apresenta o inconveniente de esta segurança ser mantida em conexão apenas ponto a ponto, sendo que para mensagens do tipo *broadcast* não existe nem um tipo de checagem de segurança.

3 DESENVOLVIMENTO

Este capitulo irá mostrar o desenvolvimento do trabalho, mostrando as técnicas utilizadas e o como foram feitos os diversos canais de comunicação necessários para se atingir o objetivo do projeto que é um sistema de automação residencial de baixo custo. O capítulo está organizado da seguinte forma: inicialmente uma explicação sobre a topologia geral do trabalho; como foi feita a comunicação de um usuário remoto com a casa; a comunicação interna da casa, via *bluetooth* e rádio frequência criptografada; o programa de supervisão desenvolvido para Android; e os dispositivos criados para teste da rede proposta.

3.1 Topologia Geral

Nesta seção será mostrada a topologia geral da rede de comunicação proposta para a interface do usuário com os controladores instalados na casa. O sistema de automação residencial proposto no trabalho pode ser dividido em dois níveis de comunicação, um utilizado para a comunicação de um usuário que está fora da casa e pretende controla-la remotamente via Internet e outro que leva os dados desde o ponto de comunicação com a Internet dentro da casa até o controlador desejado.

A comunicação entre um usuário remoto e a central de automação da casa é feita via Internet e utiliza um servidor remoto o qual é acessado tanto pelo aplicativo utilizado pelo usuário quanto pelo aplicativo da central. A Figura 7 mostra o diagrama da topologia.

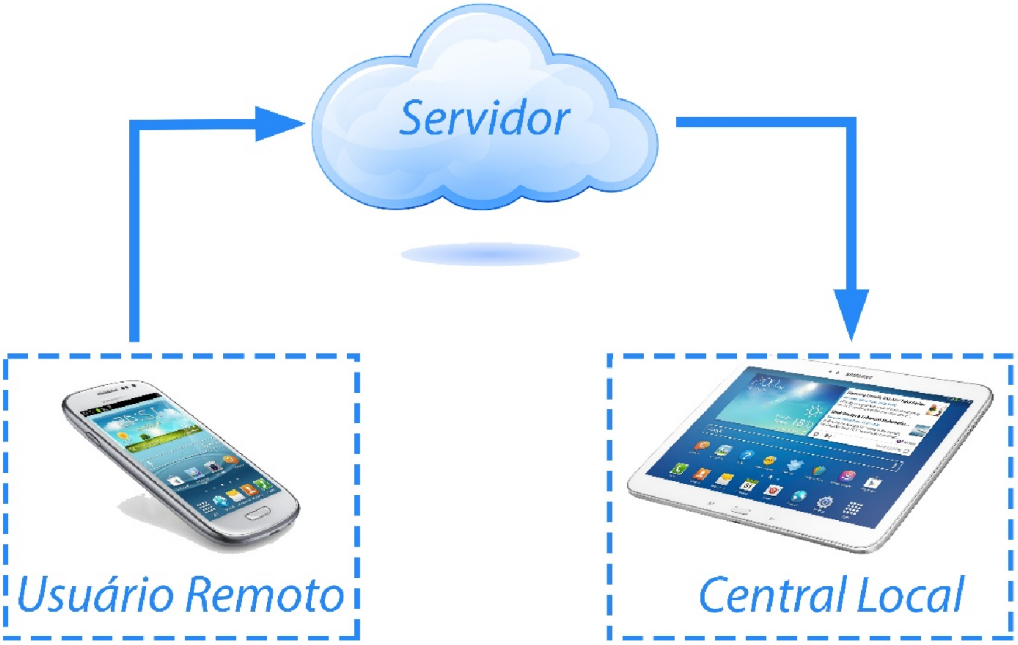

Figura 7 - Comunicação entre Usuário Remoto e a Central Local

Devido à restrição imposta no trabalho de um sistema com baixo custo é necessário o uso de tecnologias que impliquem na redução de custo em relação a um sistema de automação convencional. Por este motivo foi definida para o trabalho uma topologia com controladores utilizando comunicação sem fio com a Central Local. O uso deste tipo de comunicação reduz os custos com cabeamento e a mão de obra associada a este tipo de instalação que podem corresponder por até cerca de 45% dos custos associados a uma instalação de automação predial, quando esta é feita em paralelo a construção do prédio e até 75% do valor quando considerado um prédio já construído. (RODRIGUES, CARDEIRA e CALADO, 2010) (WANG, P. LYNCH e LAW, 2005).

Para uma redução no consumo do circuito de comunicação dos controladores e uma redução no custo dos módulos de comunicação sem fio foram utilizados dispositivos de baixa potência. Estes dispositivos possuem limitado alcance sendo necessária construir uma rede para roteamento das mensagens limitando a distância de "salto" que cada mensagem necessita percorrer, sem interferir na distância máxima que um controlador pode estar da central. A topologia escolhida para a comunicação interna está mostrada na Figura 8.

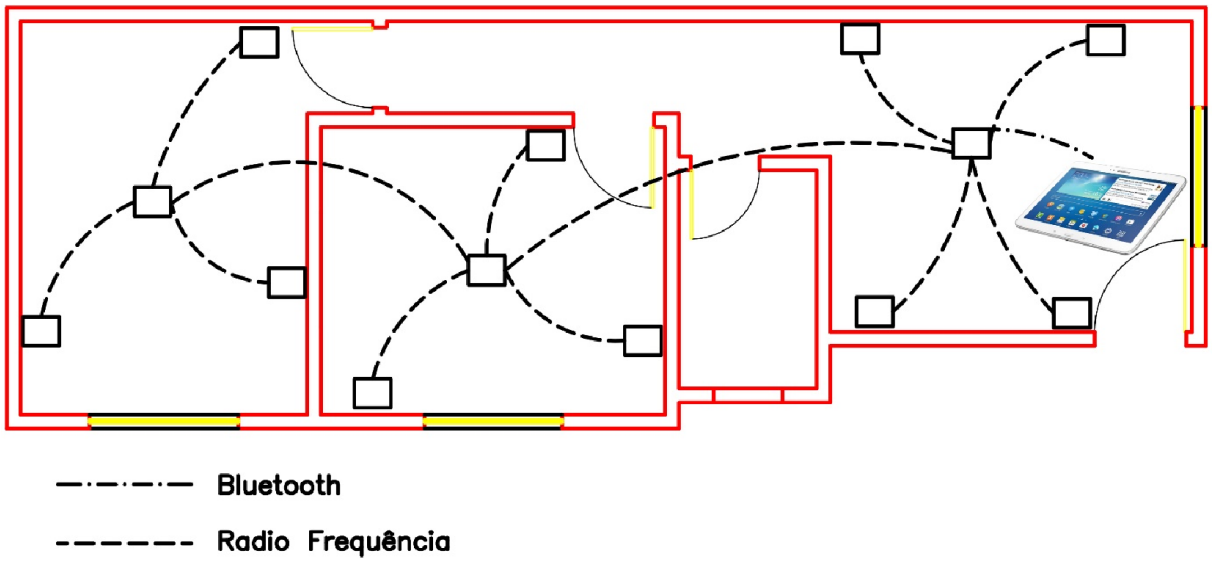

Figura 8 - Comunicação Local

A interface com o usuário, tanto na central quanto o dispositivo remoto, é feita por meio de um celular/*tablet* com Android. No controle dos dispositivos controladores foram utilizados Arduinos devido a seu baixo custo e simplicidade de programação. Como a Central Local é um *tablet* e estes dispositivos não fornecem o tipo de comunicação escolhida para a rede é necessário que este comunique-se com um Arduino que repassará a mensagem ao restante da rede de controladores.

O protocolo disponível para a comunicação escolhido entre o *tablet* e o Arduino Central foi o *bluetooth*. Esta tecnologia já tinha se mostrado eficiente no trabalho correlato de (LEE e CHOI, 2003) quando é necessária a comunicação apenas "ponto a ponto", com um dispositivo fixo e distância relativamente pequena, então essa tornou-se a escolha natural de protocolo entre as duas possibilidades envolvidas (a outra possibilidade era o *wi-fi*). Porém o sistema objetivado é de baixo custo e necessita de roteamento então deixa de ser interessante que cada controlador possua um módulo *bluetooth.* Isso ocorre porque estes controladores possuem necessidade de serem "pareados" com um dispositivo por vez impossibilitando o fácil roteamento. Outro motivo é que ainda que os módulos *bluetooth* possuam baixo custo quando comparados a tecnologia *wi-fi,* estes não são tão baratos quanto os módulos de RF, assim definiu-se que haveria um Arduino Mestre comunicando-se via *bluetooth* com a Central Local. O Arduino Mestre comunica-se então com os demais controladores via *transceivers* RF.

A topologia proposta na Figura 8 é básica, mas devido à modularidade da rede poderiam ser adicionados mais controladores para acionar mais pontos de luz ou ainda serem adicionados controladores com relés para acionamento de uma carga genérica.

A diferença necessária para montar os *layouts* mais simples e mais complexos é a adição de mais controladores. Desta forma é perceptível que a rede possui modularidade, sendo necessária uma variação do programa da IHM para ser possível controlar um número diversificado de tipos controladores.

3.2 Comunicação entre Usuário Remoto e Central Local

A comunicação entre um usuário que está fora da casa ou não está utilizando a central para atuar sobre a casa é feita via Internet, com o uso de um servidor remoto. A opção pelo uso de um servidor na "nuvem" foi feita para eliminar um problema existente com as conexões a Internet que não fornecem IP fixo. Este fato torna impossível que um dispositivo remoto conecte-se a um servidor sem o uso de algum outro artificio, como por exemplo, um gestor de DNS. Por não utilizar um servidor local o usuário também fica livre do investimento inicial para a montagem de um e também dos custos de manutenção relacionados ao servidor.

No trabalho foi escolhido utilizar uma plataforma para aplicativos *online* disponibilizada pelo Google, o *Appengine.* Este recurso disponibiliza para o usuário uma CPU para processamento e um banco de dados *online*. Na versão gratuita desta plataforma é possível armazenar até 1GB de informação e fazer 50 mil requisições de leitura ou escrita por dia.

A conexão entre cliente e servidor é protegida por uma criptografia do tipo SSL (*Secure Socket Layer*) e o formato que as mensagens devem ser enviadas pode ser alterado utilizando um algoritmo que está rodando no servidor.

Como o usuário remoto não consegue escrever diretamente na central local o comando é gravado no banco de dados do servidor. O banco de dados do servidor possui uma variável que é lida pela central com uma frequência definida. Ao ser detectada uma mudança nesta variável o central sabe que algum comando foi dado e então passa a identificar o comando e retransmiti-lo para o Arduino Central que irá encaminhar até o controlador desejado.

Para eliminar um trafego desnecessário de dados, causado pela impossibilidade de o servidor fazer uma requisição ao cliente quando detecta-se a mudança em uma variável, foi criada uma variável de *status* que indica que área/áreas sofreram mudanças. Assim a central local faz uma varredura cíclica apenas em uma variável. Ao ser detectada a mudança nessa variável a central então identifica que bit foi ativado e baseado neste bit sabe que outra variável tem que ser lida para ter a informação do comando a ser dado. Se houverem mais de um comando dados entre uma janela de leitura e a janela de leitura anterior haveriam mais de um

bit ativado e, portanto a central irá ler mais de um comando. O protocolo implementado nas mensagens entre central e usuário remoto está mais bem descrito na subseção a seguir.

3.2.1 Padrão das Mensagens

O nome das variáveis salvas no servidor é uma composição de usuário, senha e tipo da variável, logo para acessá-las é necessário o conhecimento desses três componentes. Um exemplo de variável de *status* pode ser:

Usuário: Pedro

Senha: 123456

Variável: Pedro123456_status

Os dados salvos são de dois tipos: *Status* e Ação

Dados de *Status*: Uma variável única por usuário que indica para que equipamento será direcionada a mensagem. Esta é a variável que sofre o *pooling* (varredura periódica ou em ciclos) da central local. É composta por cinco bits a cada ambiente, cada um representando um na respectiva ordem:

- Televisor;
- Receptor de TV;
- Controle de Iluminação;
- Controle de Temperatura;
- Controle de Carga.

Com essa variável é possível determinar para que ambiente e equipamento deste ambiente o comando deve ser enviado, com essa variável também pode-se determinar qual outra variável deve ser lida para determinar a ação do equipamento. Um exemplo de uso desta variável pode ser:

Ação: Controlador da Lâmpada do Ambiente 3

Variável: Pedro123456_status

Valor da variável: 00000|00000|00100|00...

Dados de Ação: Cada tipo de equipamento possui uma variável de ação. Esta é responsável por indicar que comando vai ser dado ao controlador. Para sistemas mais simples como iluminação os comandos possíveis são apenas para ligar e desligar lâmpadas porém para

o controle remoto de televisores, por exemplo, estes dados de ação tornam-se bem mais complexos.

Exemplo da Troca de Mensagens:

Um usuário remoto escreve no servidor um comando destinado a ligar as luzes na sala da casa (ambiente 1). Portanto o servidor agora possui os valores da variável Usuário+Senha_status = 00100|00000|000... . Está troca de mensagens está representada na Figura 9.

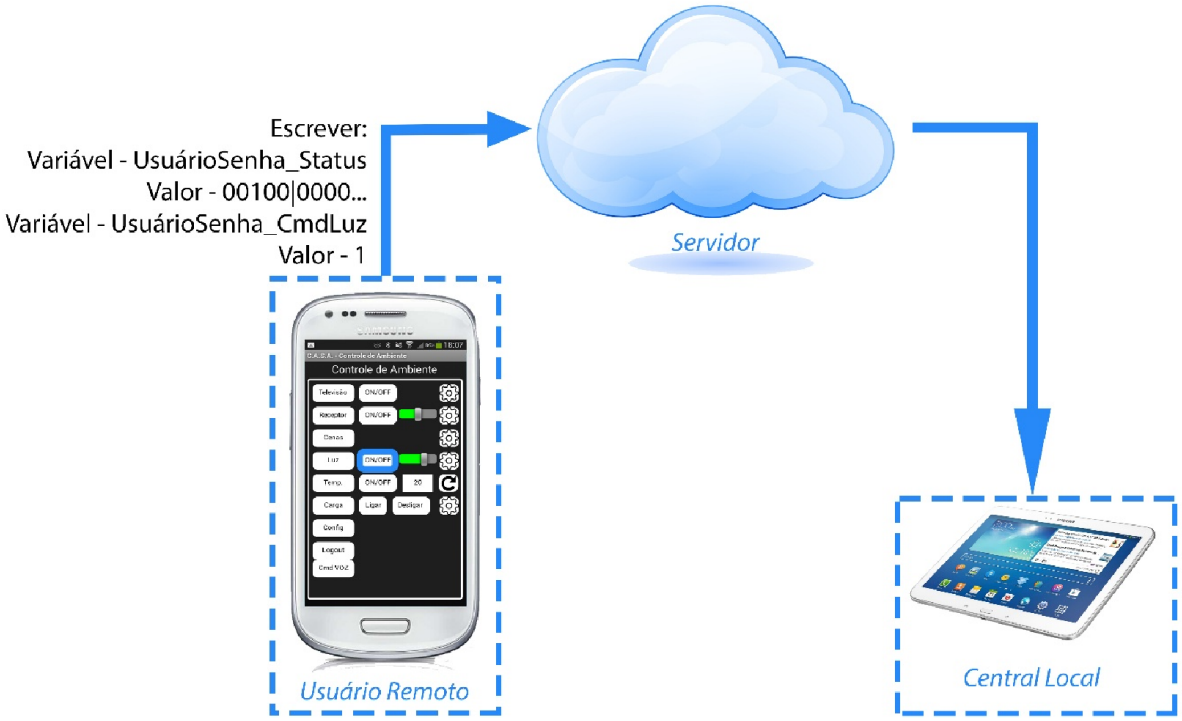

Figura 9 - Troca de mensagem entre Usuário Remoto e Servidor.

Depois que a variável é escrita no servidor ela fica aguardando para ser lida pela central local. Este tempo de espera será de no máximo um ciclo de *pooling*. Após detectar que existe um comando a ser enviado para o sistema de iluminação da sala a central local então faz a leitura da variável de comando de iluminação. Este procedimento está mostrado na Figura 10.

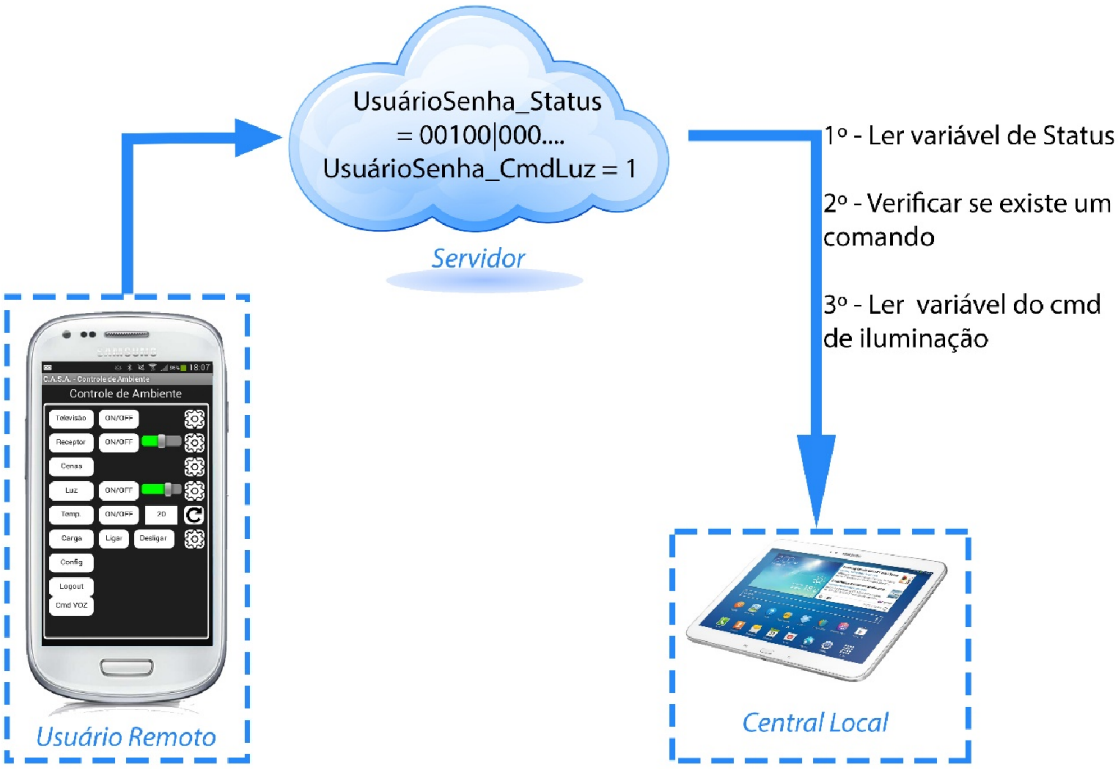

Figura 10 - Troca de mensagens entre Servidor e Central Local.

Ao término do recebimento do comando de iluminação a ser executado a central local então repassa via *bluetooth* para o Arduino central a mensagem de que a lâmpada da sala deve executar o comando de ligar. O Arduino então encaminha a mensagem até o destino. Para evitar comandos duplicados a Central Local zera as variáveis lidas do servidor, isso impede que em um próximo ciclo de leitura um comando já executado seja confundido com um comando ainda não executado. A organização da mensagem *bluetooth* e do roteamento do comando até o controlador de interesse serão explicados nas próximas seções.

3.3 Comunicação via Bluetooth

Nessa seção será explicada como se dá a comunicação entre a Central Local e o Arduino central, incluindo que módulos são utilizados e como é a formatação dos comandos passados. Todo o protocolo *Bluetooth* (ver Apêndice A) no lado do Arduino foi implementado via um módulo físico ou seja o protocolo é encapsulado em *hardware*. Isto permite que o microcontrolador apenas precise informar o pacote de dados via uma comunicação serial RX/TX, reduzindo o esforço computacional e permitindo assim o uso de um microcontrolador mais barato.

O módulo *Bluetooth* utilizado foi o HC-06. Nele é necessária a configuração de uma senha (PIN) e o nome do dispositivo. Estes procedimentos de padronização são feitos através de comandos AT enviados pela porta serial disponível no módulo. Ainda é necessário definir se o módulo é cliente ou servidor. Por padrão foi definido que o módulo seria servidor e que portanto as requisições seriam feitas pela IHM (cliente). No Apêndice B existem mais detalhes de como foi feita a parametrização dos módulos e dos comandos AT utilizados para isso. O alcance destes módulos é de cerca de 30 a 40 metros (GUANGZHOU HC INFORMATION TECHNOLOGY CO., LTD, 2010). Uma imagem do módulo está mostrada na Figura 11.

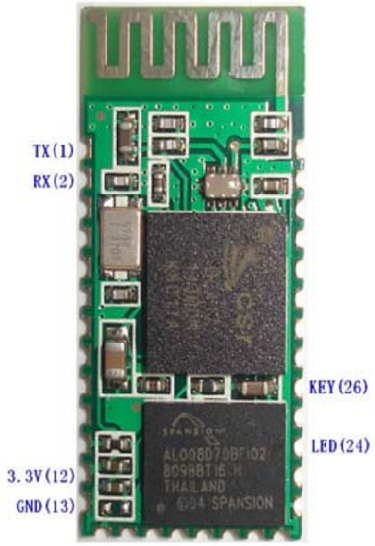

Figura 11 - Módulo HC-06
Fonte: User Guide HC Serial Bluetooth.

As mensagens enviadas entre a IHM e a central além de possuírem o protocolo *Bluetooth* tem seu pacote de dados composto por 8 bytes e está estruturado conforme a Figura 12:

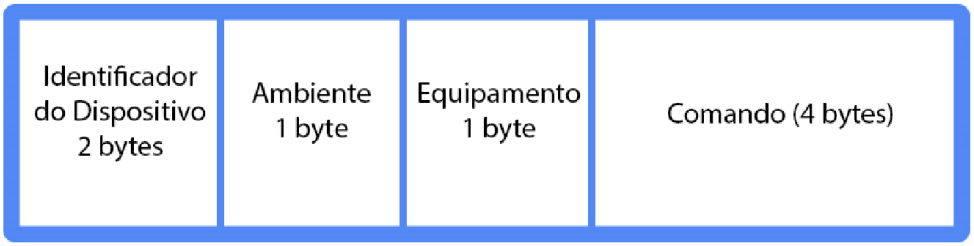

Figura 12 - Pacote de dados do *Bluetooth*

3.3.1 Identificador do Dispositivo

A parte do identificador do dispositivo é utilizada para evitar que em um caso peculiar, onde outra central que esteja no raio de alcance da Central Local e possua mesmo PIN (senha)

da central desejada, não seja pareada por engano. Cada usuário recebe o identificador do seu dispositivo e insere-o previamente no *software* da Central Local, caso o valor do identificador seja diferente do encontrado na central pareada a central informa a Central Local sobre o erro. O identificador consiste de um valor de dois bytes.

3.3.2 Ambiente

A informação sobre para que ambiente a mensagem está dirigida é importante para o correto roteamento da mensagem. Existe a possibilidade de um número muito grande de ambientes, devido a forma com que a rede de rádio frequência opera, sendo assim os *softwares* tem que sofrer uma personalização para se enquadrarem a cada usuário, ou em casos mais simples é possível usar uma numeração padrão para cada ambiente.

Foi definido por padrão uma rede com sala, dois quartos e garagem, sendo que a numeração dos ambientes começa do número 0 a partir da sala e sofre incrementos de 5 unidades a cada ambiente, assim possibilitando que cada ambiente possua 5 equipamentos.

- Sala = 0;
- Quarto1 = 5;
- Quarto2 = 10;
- Garagem = 15;

3.3.3 Equipamento

Utilizando a informação fornecida pela variável de *status* lida a partir do servidor a Central Local determina para qual equipamento será direcionada a mensagem. Por padrão o sistema permite que existam até 5 equipamentos por ambiente.

Dada a modularidade da rede e a possibilidade desta ser cascateada é viável a implementação de um ambiente com muito mais controladores do que apenas os 5 padrões porem o programa do celular restringe esse valor pois só possui interface para acionamento deste número de dispositivos, a adição de mais botões permite a expansão da rede.

3.3.4 Comando

Após a identificação de qual equipamento em que ambiente o comando deverá ser enviado é preciso saber que ação o controlador deve fazer. Sendo assim a variável lida do servidor que indica o comando para aquele determinado equipamento é quebrada em até 4 bytes (algumas mensagens de comando possuem tamanho menor) e então enviada no pacote *bluetooth* para o Arduino.

3.3.5 Exemplo de Comunicação

Para demonstrar como se dá a comunicação entre Central Local e o Arduino Central via *bluetooth* será utilizado uma continuação do exemplo da troca de mensagens mostrada nas Figura 9 e Figura 10. Neste exemplo uma mensagem foi enviada por um usuário remoto, salva no servidor, lida pela Central Local e agora será repassada até o Arduino Central conforme mostrado na Figura 13.

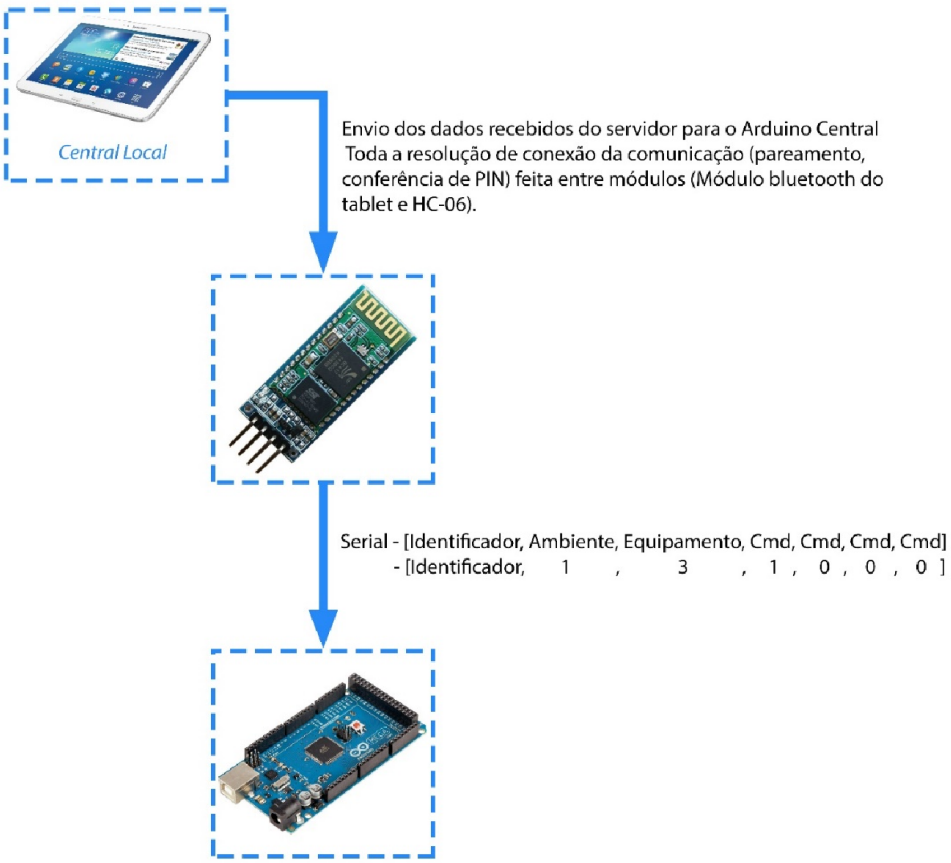

Figura 13 - Comunicação entre Central Local e Arduino Central

As mensagens recebidas pelo Arduino Central via *bluetooth* são processadas e caso o identificador não seja reconhecido estas são automaticamente descartadas. Se estiver tudo certo com o padrão da mensagem o Arduino central então tenta rotear a mensagem via comunicação rádio frequência até o controlador desejado. A forma de roteamento da mensagem será explicada na próxima seção.

3.4 Comunicação Criptografada via Rádio Frequência

A comunicação via RF é a utilizada entre o Arduino Central e os demais controladores, ela é bidirecional e com módulos que trabalham em 2.4GHz. Devido ao fato de não haver nenhum tipo de segurança no pacote de dados desta comunicação, foi adicionado um sistema de criptografia a mensagem. A escolha por criptografar o pacote de dados da mensagem com um algoritmo do tipo "senha única baseada no tempo" se dá por três motivos:

i. Segurança: No caso de um agente externo monitorar a frequência utilizada pelo módulo, com o uso até mesmo de um controlador projetado neste trabalho, e, portanto captando uma mensagem aleatória e simplesmente reproduzindo-a seria possível atuar na casa. Sendo assim o sistema não seria robusto, e pouquíssimo seguro de utilizá-lo para controle de acesso. Como deseja-se um sistema com capacidade de fazer muitas outras funções do que as citadas no trabalho incluindo segurança e abertura de portões, é interessante que o programa tenha um alto nível de proteção.

ii. Checagem de Erro: Como a mensagem é criptografada qualquer mínima alteração na mensagem resulta em uma mensagem descriptografada completamente diferente, causandoa mensagem não ser reconhecida pelo sistema e, por consequência, evitando um erro nos comandos.

iii. Interferência entre módulos: Como cada módulo recebe uma chave de criptografia única por rede, essa chave é composta por 10 bytes, evita-se o problema que já foi citado com o *bluetooth* de um módulo vizinho acabar sendo ativado por uma mensagem que não estava destinada para ele.

Os módulos são do modelo "nrf24l01+" produzidos pela Nordic com frequência 2,4 GHz e modulação GFSK. Estes módulos escolhidos são de baixa potência (até 1 mW ou 0 dBm) e baixo custo. Uma imagem dos módulos pode ser vista na Figura 14 e mais dados sobre os módulos estão no Apêndice C. A comunicação destes módulos com o microcontrolador é feito através de uma serial síncrona desenhada para periféricos de alta velocidade (interface SPI).

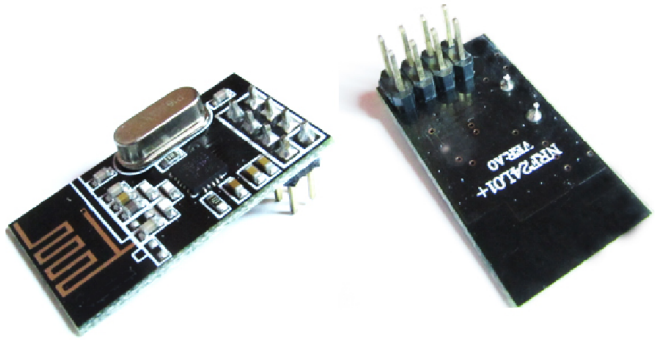

Figura 14 – Módulos de Rádio Frequência nrf24l01+

Como os módulos apresentam baixo alcance é interessante o roteamento do sinal através de uma rede, utilizando cada módulo como um ponto de retransmissão da mensagem, assim garantindo que esta possa cobrir toda a área de uma casa. O algoritmo de roteamento e a criptografia serão melhor explicadas nas próximas subseções.

3.4.1 Roteamento da Mensagem

Como já citado anteriormente os módulos escolhidos apresentam baixa potência isso implica diretamente no alcance dos módulos, que dificilmente conseguem superar os 10 metros de distância. Se considerarmos apenas 10 metros de raio ao entorno dos módulos é fácil notar que o uso destes inviabiliza a automação de uma casa inteira, permitindo apenas a automação de pequenos ambientes. Para a solução deste problema foi proposta no trabalho o uso de uma rede desenvolvida em (MANIACBUG, 2012) e testada no trabalho de (VALENTIM e MUNARO, 2014) onde também foi aplicada a um projeto de automação residencial.

O próprio criador da rede faz um paralelo entre sua tecnologia com uma largamente empregada hoje em dia, o *ZigBee*. Quando comparada ao *ZigBee* a rede utilizada neste trabalho não é tão robusta já que não permite a diversidade de topologias do *ZigBee*. Porém quando comparamos o custo é possível perceber que os módulos com "nRF2410l+" são em torno de 6 vezes mais baratos que os módulos *ZigBee*.

O algoritmo chamado de *RF24Network* propõe a implementação da camada de rede do modelo de camadas ISO/OSI. A topologia implementada é do tipo árvore ou seja apresenta a comunicação direta restrita entre "pais e filhos" sem a possibilidade da comunicação direta entre "irmãos". Com isso a única restrição para que a rede funcione em perfeita ordem é que

todos os filhos tenham comunicação com seu pai, se isso for respeitado em todas as camadas a entrega das mensagens é garantida.

3.4.1.1 Método de Roteamento

O roteamento das mensagens foi feito utilizando-se de uma técnica de roteamento por árvore, onde as nível e as relações são definidas pelo endereço de cada dispositivo. Cada dispositivo possui um endereço em formato OCTAL e as camadas são definidas por cada dígito do endereço. Por exemplo, o dispositivo pai é o 00, seus filhos são 01, 02, 03, 04, 05. Os filhos de um dispositivo da segunda camada, como o 02 por exemplo, são representador por 012, 022, 032, 042, 052. As camadas subsequentes obedecem a mesma lógica. O número 0 utilizado antes de todos os endereços é um indicador para informar que estes estão escritos na base octal.

São possíveis até 5 camadas o que permite um total de 3,125 módulos (5 módulos por camada com 5 camadas ou seja 5^5) em uma rede. Uma rede possível é mostrada na Figura 15. Caso seja de interesse o uso de mais dispositivos é possível fazer uma rede independente com os módulos operando em outro canal, já que é possível saltar entre frequências com os módulos.

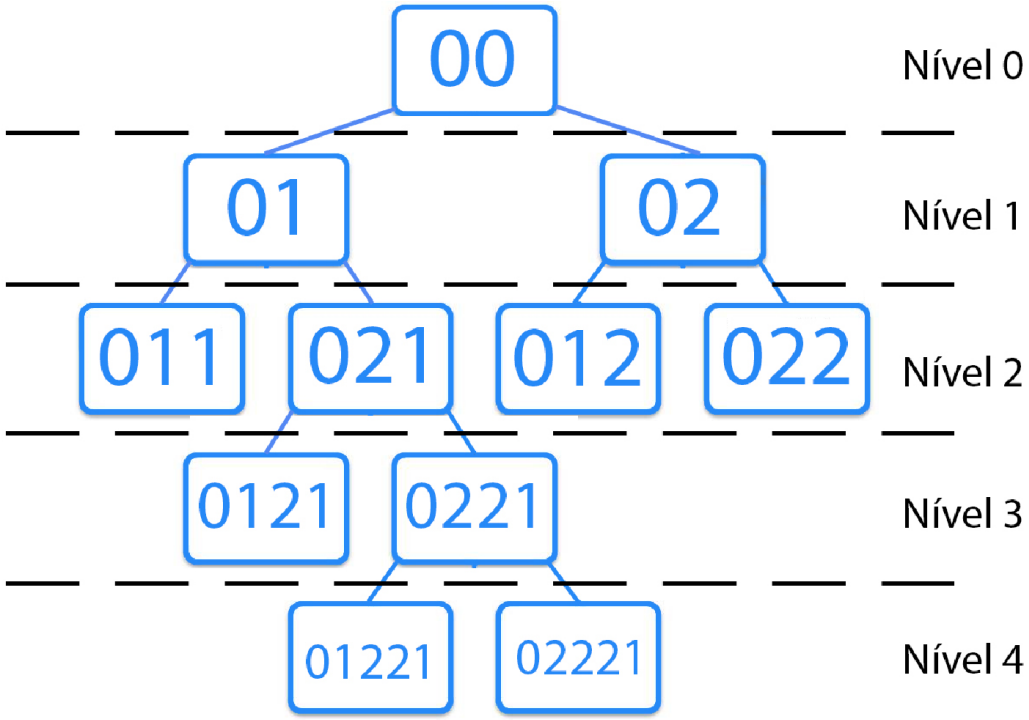

Figura 15 - Exemplo de uma rede com a topologia do trabalho

Quando uma mensagem é enviada a partir do nó 00 por exemplo e o nó de objetivo é o 0121, todos os nós que estão no alcance do 00 recebem a mensagem, porém só os nós 01 e 021 retransmitem esta mensagem,

como mostrado na

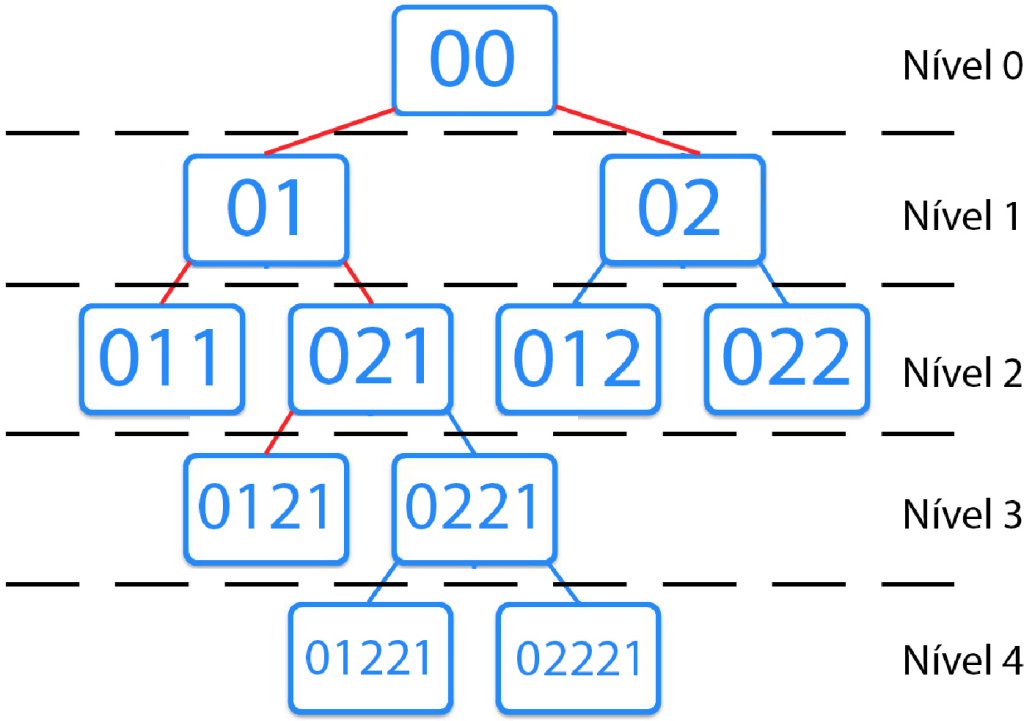

Figura 16 onde em vermelho estão as mensagens trocadas.

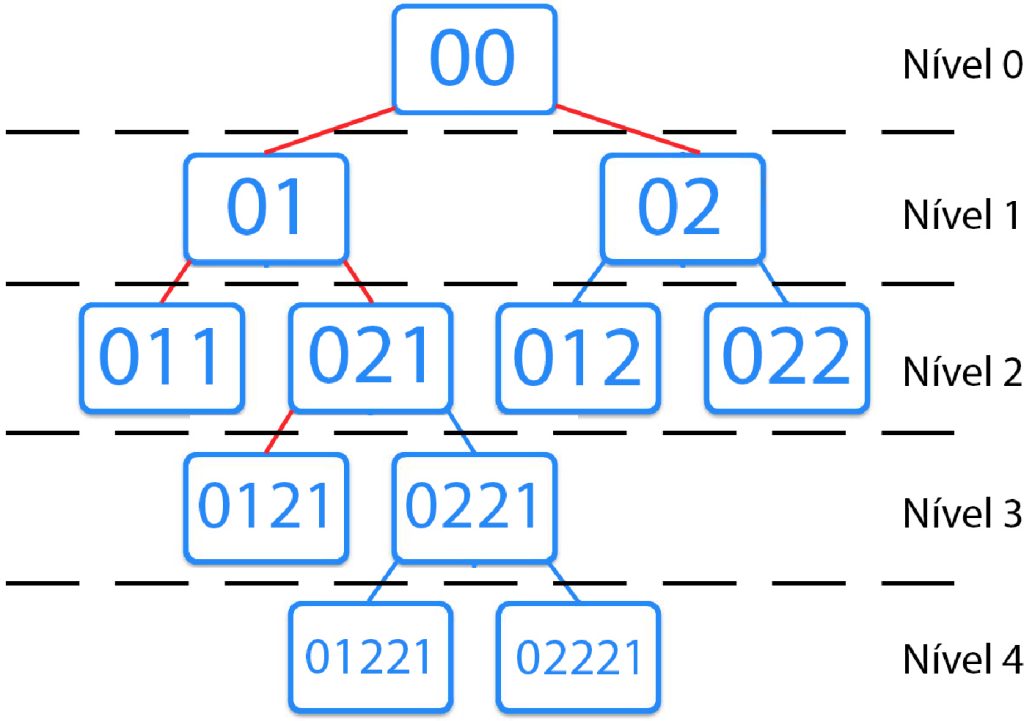

Figura 16 -Possível roteamento de mensagem

Caso dois irmãos queiram trocar uma mensagem é necessário que essa passe pelo nó pai que irá redirecionar a mensagem até o dispositivo desejado. O mesmo é feito entre dispositivos que são primos.

O algoritmo de roteamento é mostrado nos fluxograma da Figura 17 e Figura 18.

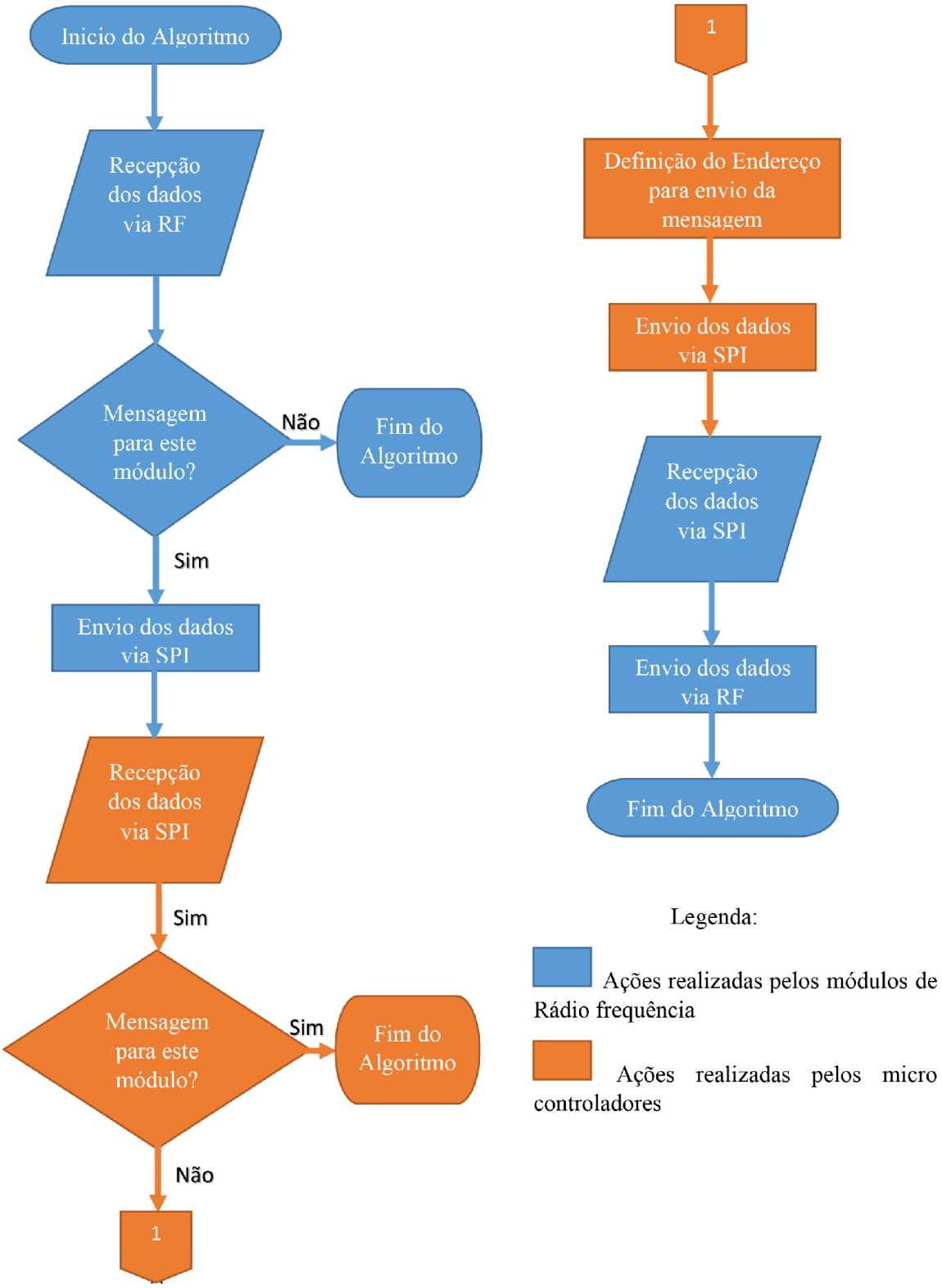

Figura 17 - Recepção e Encaminhamento de Mensagem

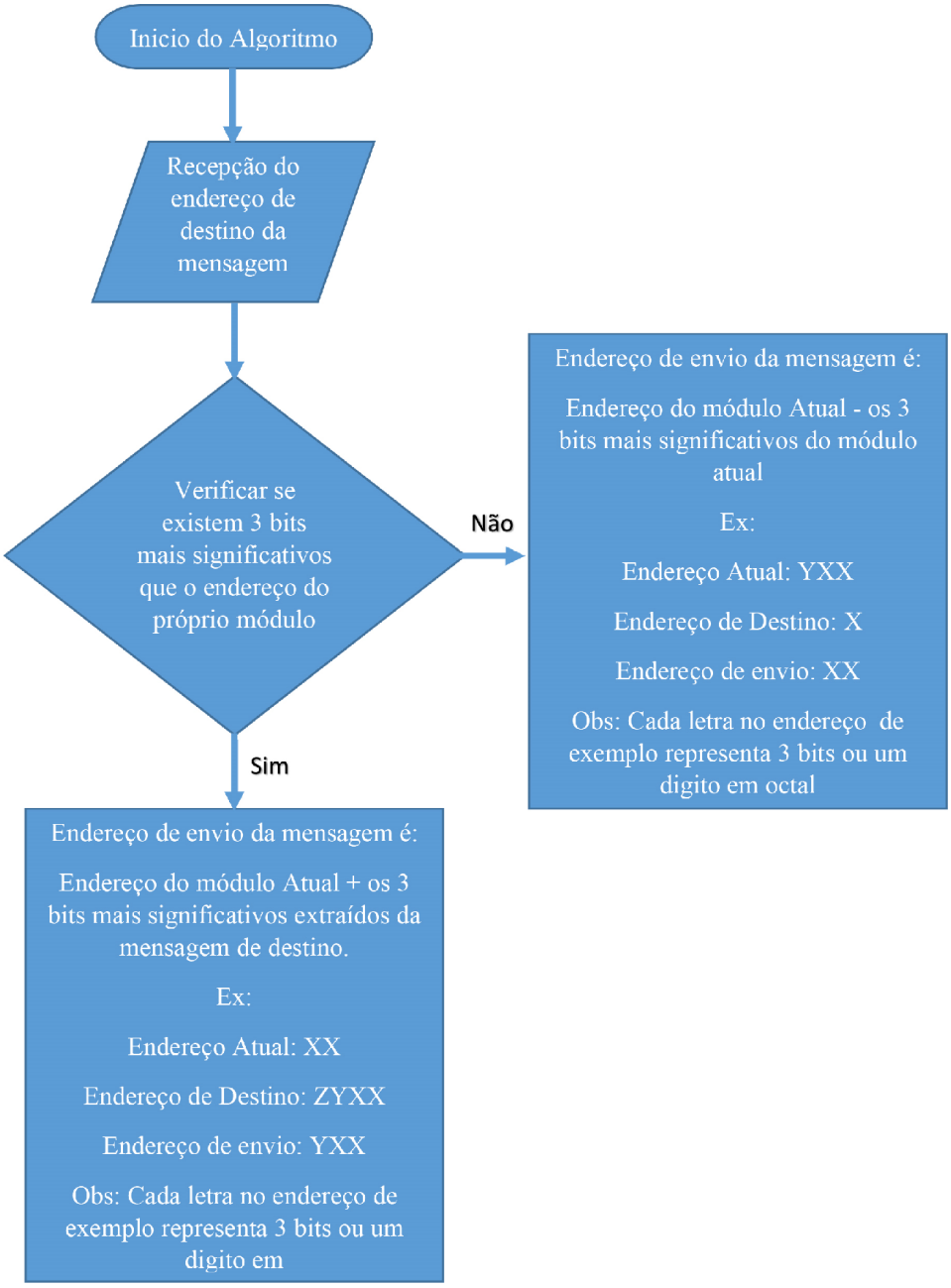

Figura 18 - Definição de endereço de encaminhamento

3.4.1.2 Alcance da rede

O alcance de uma rede de comunicação é a distância máxima na qual é possível comunicar dois dispositivos. O alcance entre cada par de dispositivos foi testado em diferentes condições que serão melhor explicadas na seção de resultados, porém considerando a possibilidade de uma barreira (parede) e o tamanho da mensagem igual ao utilizado no trabalho o alcance foi de 10 metros, onde a perda de mensagens foi reduzida a zero.

Como a rede proposta no trabalho possui até cinco níveis e cada módulo tem um alcance onidirecional de 10 metros é possível concluir que o alcance total da rede se forem utilizadas todas as camadas é de 80 metros, e que entre central e o escravo mais distante é de até 40 metros, como mostrado na Figura 19.

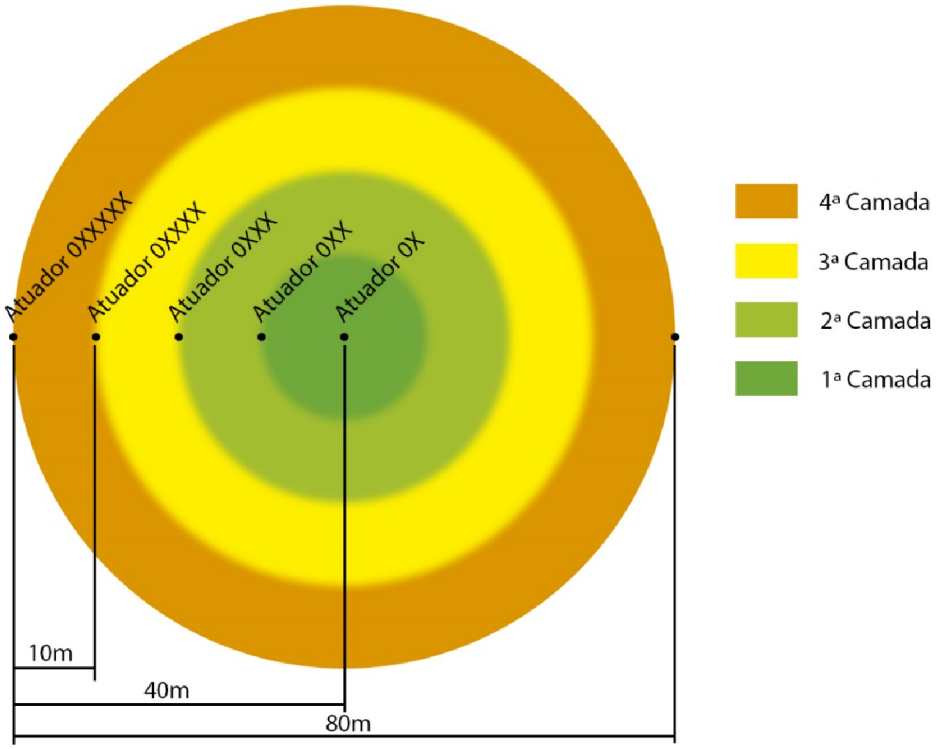

Figura 19 - Alcance da Rede

3.4.2 Criptografia

Como já explicado anteriormente o uso de criptografia na comunicação de rádio frequência vem para solucionar o principal problema inerente a redes sem fio, que é o fato do dado não estar fisicamente protegido. Sendo assim é necessária uma outra medida para garantir a integridade da informação.

Criptografia é o ato de tentar cifrar a informação com base em uma chave, assim garantindo que só aqueles que possuem a chave podem ter acesso aos dados da mensagem. É possível dividir a criptografia do trabalho em duas partes: geração de uma chave única variável no tempo e o processo de encriptação dos dados. As seções a seguir mostram como isso foi feito no trabalho.

3.4.2.1 Senha Única Baseada no Tempo

Para a chave da encriptação é utilizado o algoritmo *Time-based One-Time Password* (TOTP) (ver Apêndice D) que é um código padronizado pela RFC6238 que foi implementado no Arduino. O objetivo é gerar um código diferente a cada 30 segundos e que estes códigos sejam gerados baseados na data, hora e uma chave dos controladores e do servidor, por isso é necessário os comandos de sincronização de hora no mestre e no controlador, e também que os dispositivos da mesma rede compartilhem uma chave secreta.

Partindo do pressuposto que a data esteja sincronizada com um erro de mais ou menos 15 segundos temos então a mesma senha gerada no mestre e no controlador. A data é sincronizada na inicialização dos módulos com o uso de uma comunicação via RF que não possui criptografia. Essa senha só é valida por 30 segundos e é uma senha nunca se repete antes do ano de 2038, garantido que alguém que escute a frequência de comunicação acabe por reproduzir uma mensagem invalidada pela criptografia.

3.4.2.2 Encriptação

Utilizando a senha gerada pelo algoritmo TOTP é feita uma criptografia do tipo *Data Encryption Standard* (DES) (ver Apêndice E). A escolha do DES se deve ao microcontrolador utilizado ser de 8*bits* o que torna o uso de outros métodos de criptografia como o AES que possui por padrão 128 bits para criptografia algo inviável de ser calculado de forma rápida, o DES tem cálculos com variáveis de 32 bits. Porém o DES é um algoritmo que pode ser facilmente “quebrado”, mas isto ocorre quando este possui senha fixa, no caso do trabalho a senha varia a cada 30 segundos sendo necessário assim “quebrar” o algoritmo em apenas 30 segundos. Utilizando o DES gera-se a mensagem a ser enviada pelo módulo RF, na chegada nos controladores o processo contrário é feito utilizando a senha TOTP que supõe-se que é igual dado o fato de os módulos estarem sincronizados e pertencerem a mesma rede. Caso as condições pressupostas não forem verdadeiras a mensagem descriptografada não fara sentido e será descartada, conferindo assim segurança a rede.

3.4.3 Mensagem entre módulo e microcontrolador

A mensagem é enviada via SPI (4-fios) para os módulos “nrf24l01” com uma taxa de 1 Mbps, para o envio sem fio de uma mensagem basta que sejam escritos os dados em uma posição de memória do módulo que aponta para uma estrutura do tipo FIFO (primeiro a entrar, primeiro a sair) com 32 *bytes* concatenados. Os dados podem ocupar um espaço menor que o disponível, com o tamanho do pacote sendo detectado automaticamente pelo módulo através de um terminador. Os dados são enviados assim que é escrito em outro endereço de memória o destino da mensagem. Quando a mensagem é enviada essa posição de memória que continha o endereço de destino é automaticamente “limpa” pelo módulo. Ainda é possível o uso de confirmação de recebimento, neste caso o módulo recebe a confirmação vinda do módulo de destino e a envia via SPI para o microcontrolador.

A camada de rede responsável pelo roteamento da mensagem foi implementada no cabeçalho do pacote de dados, ficando este composto por:

– Endereço de Destino da Mensagem: Indica em que controlador a mensagem deve chegar, ele é carregado através de todos os saltos da mensagem para que o microcontrolador que serve como ponte para o destino possa rotear corretamente a mensagem. O tamanho é de 3 *bytes*.

– Tipo de Mensagem: No protocolo proposto por (MANIACBUG, 2012) é possível que existam vários tipos de mensagens na rede. Por isso foi criado um cabeçalho (*header*) com 1 *byte* para identificar que tipo de mensagem está sendo enviada ou recebida e qual a estrutura dos dados contidos nela. Neste trabalho existem dois tipos de mensagem:

– Criptografada padrão: Mensagem cifrada que contém dados de ação para algum controlador ou somente um sinal de ressincronismo para o relógio de cada módulo.

– Sincronismo: Mensagem não cifrada utilizada para fazer o sincronismo inicial dos relógios dos controladores. Isto se faz necessário pois a senha é baseada no tempo e os módulos não possuem outra entrada de dados além da feita pela rede sem fio.

– Dados: São os comandos propriamente ditos e tem tamanho padrão de 8 *bytes*

Sempre que uma mensagem vai ser enviada um algoritmo rodando no microcontrolador identifica para que camada a mensagem deve ir e em que dispositivo ela está. Por exemplo, estamos na camada um, no dispositivo 01 e é desejado o envio de uma mensagem para o dispositivo 0121. O programa identifica que o endereço do próximo módulo a receber a mensagem deve ser 021 e então envia a mensagem para este módulo. O módulo 021 recebe a mensagem, abre o *payload* onde vai estar contido o endereço de destino da mensagem identifica

para onde enviar e reenvia a mensagem. Caso ele próprio seja o destino da mensagem os reenvios cessam e ele decriptografa o restante do *payload*, se necessário, tendo assim o comando útil para o controlador.

3.5 Programa de Supervisão

Como já citado anteriormente foram desenvolvidos dois aplicativos para o sistema operacional Android, transformando um celular ou *tablet* em uma interface para controle do sistema. Para a construção dos programas foi utilizada uma plataforma disponibilizada pelo MIT(Massachusetts *Institute of Technology*) chamada de *APPinventor*[15], onde é possível fazer uma programação de alto nível em uma interface gráfica com diagrama de blocos.

Está seção do trabalho vai introduzir a ferramenta utilizada na programação, a interface gráfica do programa, suas duas vertentes e a forma de operação do programa.

3.5.1 APP Inventor

O *App inventor* é uma plataforma para desenvolvimento de aplicativos para Android desenvolvido pelo Google e atualmente mantido pelo Instituto de Tecnologia de Massachusetts (MIT). O programa designa um modo de designer (projeto) para a construção das telas e uma parte de edição de blocos onde é feita uma programação gráfica via blocos. O aplicativo ainda roda diretamente do navegador de Internet e armazena os dados na nuvem, diretamente em um servidor do Google.

É possível que sejam feitos os testes do aplicativo em "tempo real" (online), por simulação de um dispositivo no computador ou ainda por meio de um celular ligado a Internet. Depois de finalizado, o aplicativo pode ser compilado e instalado como qualquer outro programa em um dispositivo com Android ou ainda colocar o aplicativo na *Google Play*, loja *online* que disponibiliza aplicativos para o sistema operacional da empresa.

O *App Inventor* foi desenvolvido no sentido de permitir que todas as ferramentas disponíveis em dispositivos com Android estivessem ao alcance dos programadores, sendo possível completa abstração de *hardware*, ou seja, é possível gerar um programa para rodar em diferentes aparelhos sem preocupação com o modelo especifico do equipamento. Recursos avançados, que não são padrão da plataforma Android e que são disponibilizados por alguns equipamentos, não podem ser utilizados por meio do *App Inventor*.

[15] – Site do projeto *APPinventor:* http://ai2.appinventor.mit.edu/

3.5.2 C.A.S.A.

A interface gráfica do programa desenvolvido para o projeto é em sua maioria comum tanto para a Central Local quanto para a usada pelo Usuário Remoto, o programa foi chamado de Controle, Automação e Supervisão de Ambientes (C.A.S.A.). O programa é composto por sete telas principais, sendo elas:

- Tela Inicial
- Tela de Seleção de Ambientes
- Tela de Configuração
- Tela de Operação
- Tela do Controle da Televisão/Receptor de TV
- Tela de Iluminação
- Tela do Ar condicionado

Estas telas citadas acima estão disponíveis tanto para o aplicativo que roda na central local (*tablet*) quanto para o aplicativo que é utilizado por um usuário remoto em seu celular. Existem ainda telas auxiliares, como as de digitação de canal, uma para inserção do código da rede e outra para selecionar qual dispositivo *bluetooth* o sistema deve se conectar.

3.5.2.1 Central Local vs. Usuário Remoto

Nesta subseção serão explicadas as diferenças básicas quanto aos aplicativos que rodam na Central Local em relação ao aplicativo utilizado por um Usuário Remoto. A tabela a seguir mostra resumidamente estas diferenças.

Tabela 4 - Características dos aplicativos

	Central Local	**Usuário Remoto**
Dependência da Internet para efetuar comandos		X
Necessário para acionamento remoto	X	X
Comunicação Bluetooth	X	
Telas principais	X	X
Configuração da rede de sensores	X	
Checagem de senha com o servidor	X	X

Como pode ser observado na Tabela 4 o aplicativo utilizado por um usuário remoto é obrigado a ter conexão à Internet para enviar os comandos, mesmo que este aplicativo não esteja operando de forma remota (esteja no alcance da central via *bluetooth*). Isso se dá porque a comunicação *bluetooth* permite apenas uma conexão por vez, ficando esta disponível apenas para a Central Local, assim evitando colisão entre tentativas de conexão.

Todas as telas para comando estão disponíveis em ambas as versões do aplicativo, porém as telas ligadas a configuração da rede (escolha do módulo *bluetooth* e código da rede) só podem ser acessadas via a versão para usuário local.

A senha de acesso a Central Local é conferida com o servidor, isso garante a segurança do sistema já que é impossível dar comandos locais sem a verificação do usuário. Porém é necessário que no momento do *login* no aplicativo da Central Local o dispositivo possua acesso à Internet. O Usuário Remoto tem naturalmente seu controle de acesso regulado pelo servidor via checagem de senha.

3.5.2.2 Interface Gráfica e Funcionamento do Aplicativo

Nesta subseção será mostrada a interface gráfica desenvolvida para o aplicativo e fluxogramas de navegação nas telas. Juntamente com as telas serão explicados alguns dos recursos implementados no sistema e como estes funcionam.

Em todos os fluxogramas será omitida a forma de comunicação entre o aplicativo e o Arduino Central, sendo esta genericamente chamada de “comunicação com Arduino Central”. Isso se dá pelo fato de que o Usuário Remoto apresenta um caminho diferente do seguido pela Central Local para a entrega dos comandos. Como este processo já foi explicado nas seções anteriores faz-se desnecessário a repetição desses passos.

A única exceção de diagrama que será mostrado especificamente o da Central Local é o da Figura 20, pois é necessário para explicar a forma de inserção do código da Central Local e o endereço desta.

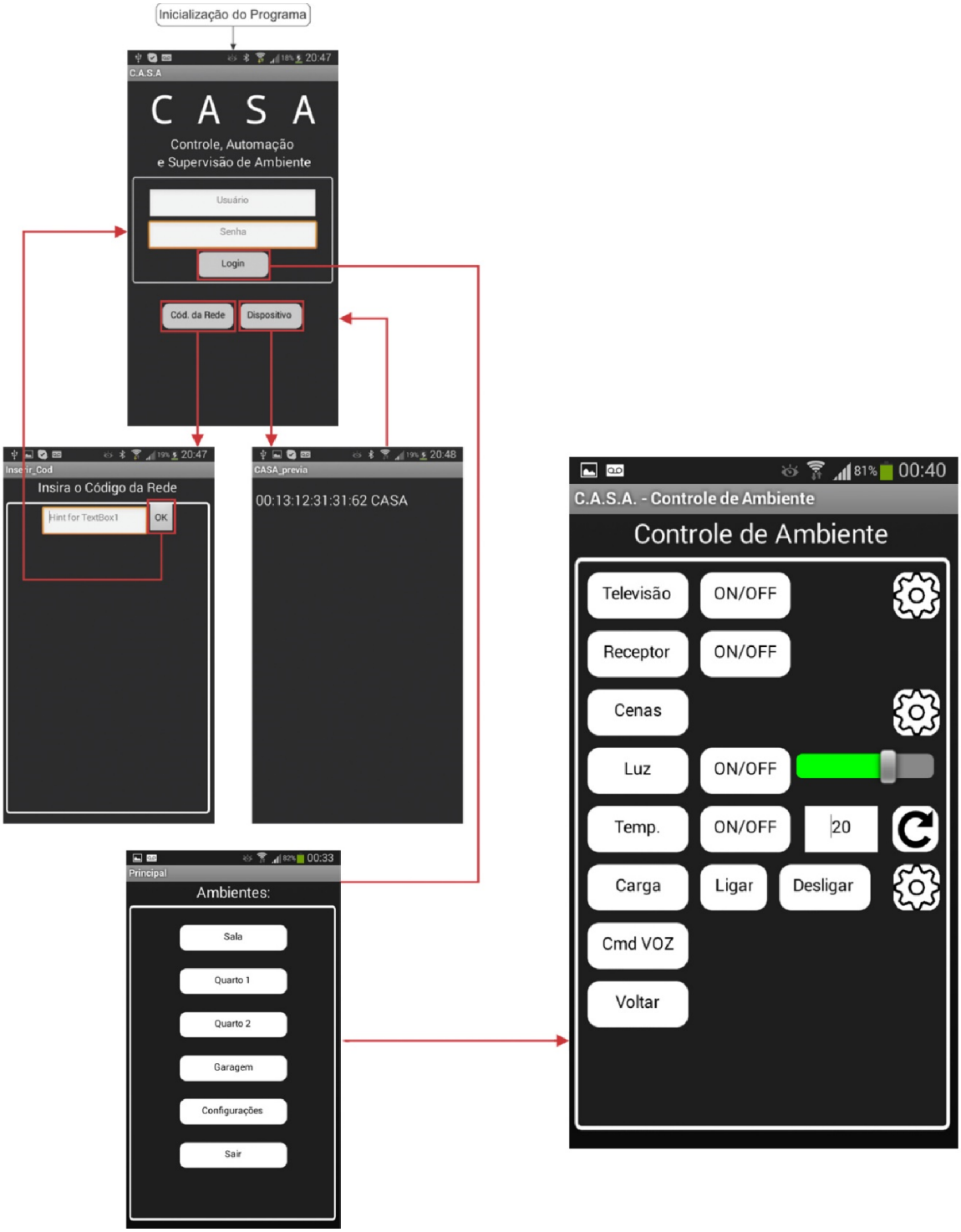

Figura 20 - Fluxograma de Telas de Inicialização

Na Figura 20 é possível ver um fluxograma da inicialização do programa, ele mostra que na primeira inicialização do programa ainda não existe uma definição do identificador da rede e nem de qual dispositivo o sistema deve se conectar, supondo que existam outros dispositivos *bluetooth* ao redor do aparelho. Após as definições na primeira inicialização isso

deixa de ser necessário, devido ao fato de que essas definições são guardados em uma ferramenta de banco de dados chamada de *TinyDB*[16].

Com as definições feitas é inserido o usuário e senha para a tentativa de *login*, por padrão o usuário e senha são definidos junto ao servidor. Caso o usuário e/ou senha tenham sido digitados erroneamente surge uma mensagem de erro dizendo que estes não conferem. Esse processo de conferencia é feito através do envio tanto de usuário quanto da senha para o servidor que responde o identificador do dispositivo ou um código de erro. Caso seja recebido um identificador diferente do informado ou um código de erro é mostrado na tela que o *login* não foi concluído com sucesso e o dispositivo não se conecta a central.

Existe também a possibilidade de não haver um dispositivo definido ou estar fora do alcance da central, caso isso ocorra uma mensagem de erro também informa o usuário de que não foi possível efetuar o *login*.

Caso a conectividade *bluetooth* do aparelho esteja desativada, na inicialização do programa é exibida uma mensagem para habilitá-la ou sair do programa. Caso o *bluetooth* seja desabilitado na tela de inicialização quando for feita a tentativa de *login* ele informará sobre o erro.

Uma imagem com a compilação destes avisos de erro pode ser observada na Figura 21.

Figura 21 - Mensagens de Erro

[16] – O *TinyDB* é um pequeno banco de dados que é inicializado quando aplicativos criados com o *App inventor* são inicializados. O banco de dados é importante porque se um aplicativo define o valor de uma variável e, em seguida, o usuário fecha o aplicativo, o valor dessa variável não será lembrado na próxima vez que o aplicativo é executado. Em contraste, *TinyDB* é um armazenamento de dados persistente para o aplicativo, ou seja, os dados armazenados não estarão disponíveis a cada vez que o aplicativo é executado.

Figura 22 - Controle da Televisão

Na tela de operação estão posicionados a maior quantidade de itens de controle, dela é possível acessar qualquer tela de controle e ainda as tarefas mais simples como ligar e deligar equipamento e mudar a temperatura do ar podem ser executadas diretamente por ela.

A Figura 22 mostra o uso do controle remoto para TV, sendo que o controle para um receptor é exatamente igual.

Sempre que um botão é pressionado o programa envia o código corresponde aquele botão, esse código é repassado até o controlador que então utiliza esse código em uma função, transformando-o em pulsos que são enviados a um LED IR para controlar o televisor.

Na tela do controle existe ainda uma tecla que abre uma tela com números para a seleção do canal. Essa opção de *design* foi escolhida devido ao fato de alguns celulares possuírem telas pequenas, portanto tornando difícil de clicar no botão correto em um controle com muitos botões pequenos.

Os códigos dos controles são armazenados dentro do controlador infravermelho responsável pelo acionamento do televisor, dessa forma a comunicação não fica saturada com longos códigos. O inconveniente gerado por essa escolha de estocar os códigos dentro do microcontrolador é que assim torna-se necessário atualizar esse controlador para cada dispositivo infravermelho diferente.

Existe uma opção de desligar/ligar a televisão de forma temporizada que está disponível na tela de configuração. Quando esse recurso é ativado o controlador recebe a ordem e conta o tempo utilizando seu relógio de tempo real (RTC), após o período de tempo selecionado ele então envia o comando de ligar/desligar.

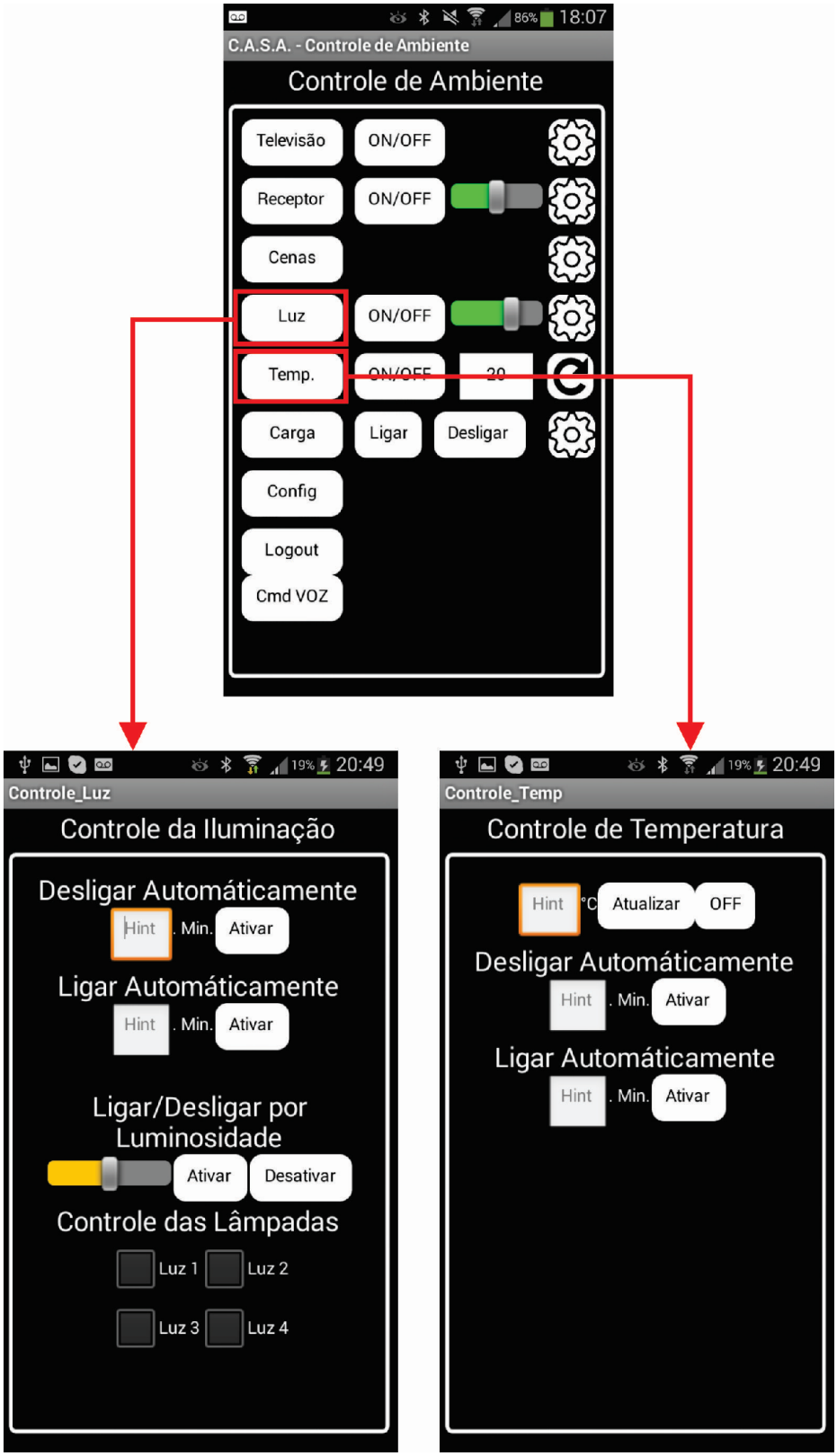

Figura 23 - Controle de Iluminação e Ar condicionado

Assim como para a televisão e receptor é possível ligar e desligar iluminação e ar-condicionado a partir da tela de operação, considerando que ainda o ar condicionado pode ter sua temperatura alterada diretamente da tela de operação.

Quando utilizado o liga/desliga para iluminação a partir do botão presente na tela de operação, todos os controladores que tem a função de controle de iluminação são ligados ou desligados. Para o controle dos pontos de luz de forma individual é necessário acessar o menu de iluminação clicando sobre o ícone *Luz*. Nesta tela existe, assim como para a televisão, um temporizador para as funções de ligar e desligar.

Além disso, como já citado existe um controle individual dos pontos de luz e ainda é possível definir que as luzes sejam ligadas pela luminosidade do ambiente. Isso é feito através de um sensor (fotoresistor) instalado nos controladores, que informa para cada um a luminosidade naquele ponto. Cada controlador liga assim que a luminosidade fica abaixo de um determinado ponto e desliga quando está se eleva acima de 40% do valor determinado. Deve haver um cuidado para que a luz não incida diretamente sobre o sensor, pois caso contrário a cada vez que a lâmpada fosse ligada o sensor teria um aumento abrupto da iluminância e portanto desligaria a lâmpada.

A tela de operação ainda permite o controle de uma carga qualquer que está conectada a um controlador com um relé. Essa função é extremamente simples e apenas liga e desliga o relé quando os respectivos comandos são enviados.

O botão de *Config* leva a uma tela onde existe um botão para enviar a central e aos controladores a data correta assim sincronizando toda a rede. A central pode ser sincronizada a qualquer momento, porém os controladores só podem ser sincronizados no momento em que são reinicializados. Após a reinicialização os controladores ficam então aguardando a data e uma nova chave de criptografia enviada pela central. Na tela *Config* também é possível alterar nome e senha do usuário.

A primeira comunicação dos controladores com a central é livre de criptografia devido ao fato de os controladores não estarem sincronizados e desconhecerem a chave de criptografia da rede. Após o sincronismo inicial não é mais possível, sem reiniciar, mudar a chave ou a data do módulo, que passa a ter ajustes periódicos e automáticos feitos pela central.

O aplicativo também faz uso do reconhecimento de voz para permitir que o usuário controle o ambiente apenas falando, para acionar o comando de voz é necessário apertar o botão “Cmd VOZ”.

3.6 Arduino Central

O Arduino Central é o dispositivo responsável por se comunicar com a Central Local e distribuir os sinais para os controladores. Ele é composto por um Arduino MEGA 2560, um módulo *bluetooth* HC-06, um módulo de rádio frequência nrf24l01 e uma fonte de alimentação. Este dispositivo ainda pode fazer a função de algum controlador, apenas com a adição de um relé ou um LED IR. A forma de interconexão dos componentes do Arduino Central está mostrada na Figura 24.

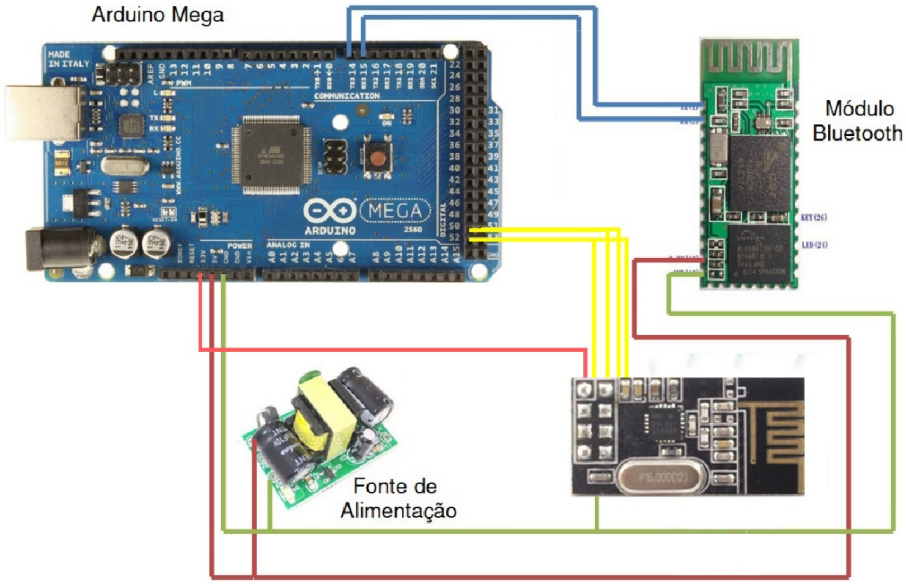

Figura 24 - Topologia da Central

Foi utilizado um Arduino MEGA 2560 para a central, pois este apresenta mais recursos de processamento em comparação com os utilizados nos controladores. Para a central são necessários mais temporizadores e mais memória RAM e ROM já que está central pode reunir funções de controladores além das próprias funções da central. Uma imagem de um Arduino MEGA pode ser vista na Figura 25.

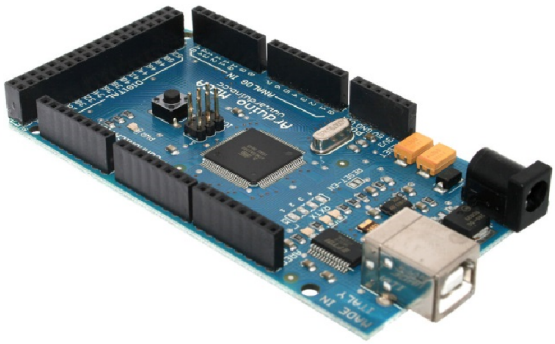

Figura 25 - Arduino MEGA 2560.

A fonte de alimentação utilizada é um circuito eletrônico do tipo *flyback* com um retificador meia ponte em sua entrada permitindo assim que seja ligado diretamente a tomada. A fonte é universal, logo pode operar de 85~220V. Além disso a fonte é completamente isolada, o que valida os produtos para venda no mercado quanto a essa característica. A saída do *flyback* se dá em 7 V passando por um regulador linear para que seja entregue a carga 5 V regulados. A potência da fonte é de no máximo 3,5 W. O conversor utilizado está representado na Figura 26. Os módulos de *bluetooth* e RF já foram mostrados anteriormente nas seções 2.2 e 2.3, Figura 11 e Figura 14, respectivamente.

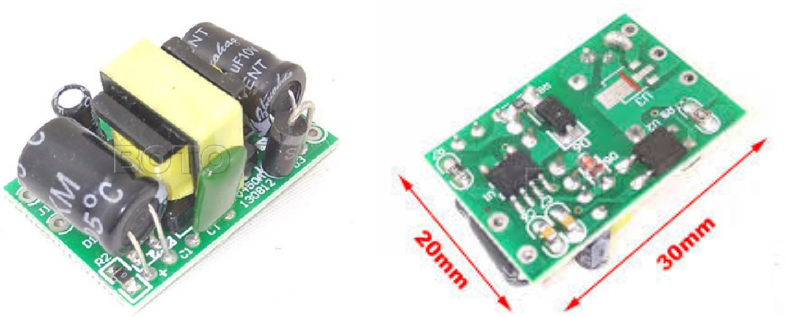

Figura 26 - Flyback com vista superior e inferior

3.7 Controladores

São os elementos responsáveis pela atuação nos ambientes, existe um tipo para cada função, neste projeto foram implementados apenas 3 tipos. Um para controle de iluminação que serve também para acionamento genérico de cargas do tipo *on/off*. Um para o controle de televisores e receptores e outro para controlar o ar-condicionado. A diferença entre os de televisão e os de ar-condicionado é meramente ligada a *software*. Uma topologia genérica dos controladores é mostrada na Figura 27.

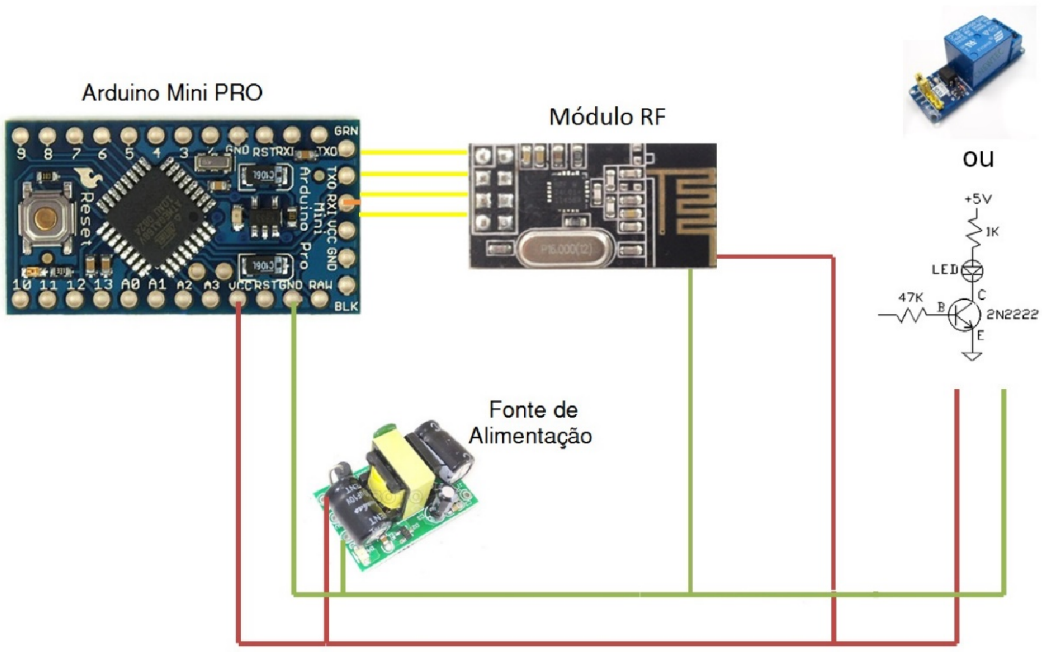

Figura 27 - Topologia dos Controladores

3.8 Iluminação

O controlador de controle da iluminação é composto por um Arduino Pro Mini, um relé, um receptor e uma fonte de alimentação. O uso de um Arduino Pro Mini é justificado pelo seu pequeno tamanho, sendo que este mede 17 mm x 43 mm. A Figura 28 mostra um comparativo entre o tamanho de um Arduino MEGA e um Arduino PRO Mini.

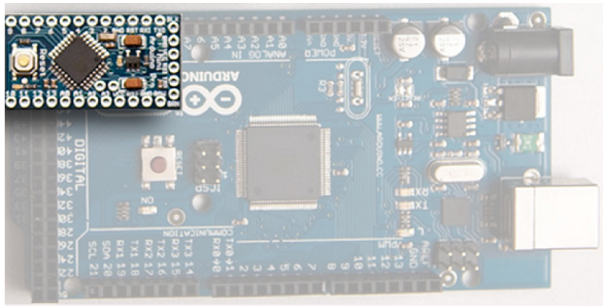

Figura 28 - Comparativo entre Arduino MEGA e Mini PRO

A fonte de alimentação é igual a utilizada no Arduino Central e já mostrada na Figura 26, o módulo de rádio frequência também já foi explicado anteriormente, na seção 2.3, onde a Figura 14 mostra uma imagem do dispositivo. O relé utilizado tem um circuito de acionamento com opto acoplador e diodo de roda livre. A placa do relé é mostrada na Figura 29.

Figura 29 - Placa de interface para Relé opto acoplada.

Para ligar um ponto de iluminação é enviado uma mensagem a partir da Central Local via *Bluetooth* até o Arduino Central. A estrutura desta mensagem é conforme anteriormente explicado na seção da comunicação *Bluetooth* e a mensagem em si é igual à da Tabela 5.

Tabela 5 - Mensagem Bluetooth para iluminação

Identificador do Dispositivo	**Ambiente**	**Equipamento**	**Comando**
0 - 65535	0 – 7	3	0 - 9
(2 bytes)	(1 byte)	(1 byte)	(4 bytes)

No comando o número de 0 a 9 que indica qual é o ponto de iluminação a ser ativado. Em um primeiro momento a interface do programa só suporta de 1 a 4 pontos de iluminação, mas a topologia de rede permite mais controladores de iluminação. Existe um modo onde é possível mandar uma mensagem para todos os dispositivos ligarem ou desligarem, isso ocorre quando é pressionado o botão *on/off* da iluminação na tela de operação, neste caso o comando é substituído por “0” ou “1”. Os valores 2 a 3 representam ligar e desligar respectivamente o ponto de iluminação 1, o 4 a 5 são utilizados para o ponto 2 e assim sucessivamente.

Quando a mensagem é recebida no Arduino Local ela é validada (conferencia do identificador) e então uma mensagem criptografada é propagada pela rede de RF, até que essa chegue ao controlador de destino, que tem o endereço calculado com base no ambiente e no equipamento de destino da mensagem.

Tabela 6 - Mensagem Bluetooth para iluminação temporizada

Identificador do Dispositivo	**Ambiente**	**Equipamento**	**Comando**	
0 - 65535	0 – 7	3	0 - 1	0 - 2^{24}
(2 bytes)	(1 byte)	(1 byte)	(1 byte)	(3 bytes)

A central recebe a mensagem e remove o tempo que é representado pelos 3 últimos bytes do comando, este tempo está em minutos. O comando de se a lâmpada deve ser ligada ou desligada é representado pelo primeiro byte do comando e é salvo na memória do controlador

de destino juntamente com a data, que é o resultado da data atual somado aos minutos de temporização. No momento que a data for igual à data atual o controlador de iluminação executa a tarefa salva na memória. Para desabilitar a temporização é enviada uma mensagem com temporização 0.

A função de ligar ou desligar por luminosidade funciona com uma mensagem *bluetooth* conforme a Tabela 7.

Tabela 7 - Mensagem Bluetooth para iluminação por Luminosidade

Identificador do Dispositivo	**Ambiente**	**Equipamento**	**Comando**	
0 - 65535	0 – 7	3	10	0 - 100
(2 bytes)	(1 byte)	(1 byte)	(1 byte)	(3 bytes)

A informação de comando é dividida no primeiro *byte* onde está presente o valor 10 que indica que o restante dos *bytes* indicara uma luminosidade e os 3 últimos *bytes* que contém um número de 3 dígitos que informa uma iluminação. O valor não tem unidade de medida e é apenas um valor relativo à calibração do sensor que informa se este está mais claro ou mais escuro. Futuramente pretende-se calibrar os sensores para corresponderem a uma iluminância em lumens.

Quando o controlador de iluminação é utilizado para o controle de uma carga genérica o *software* da central local apenas envia uma mensagem considerando que a carga é o ponto de luz 1.

3.9 Televisão e Receptor

Os controladores para televisão e receptor são constituídos de um Arduino Pro Mini, um LED IR, um módulo RF e uma fonte de alimentação. Quase todos os componentes, com exceção do LED infravermelho já foram apresentados anteriormente, sendo que a fonte, o Arduino e o módulo de RF são iguais em todos os controladores. Para os comandos de ambas as mensagens enviadas via *bluetooth* é seguido o padrão da iluminação porém o comando agora indica qual botão do controle foi pressionado.

Para o processo de temporização para ligar e desligar também é utilizado o mesmo princípio apresentado para a iluminação.

3.10 Ar-condicionado

Assim como para televisão e receptor, o controlador responsável pelo ar-condicionado é formado por um Arduino Pro Mini, um LED IR, um módulo receptor e uma fonte de alimentação. A grande diferença entre esses dois módulos é o fato de que os códigos dos controles possuírem um tamanho elevado (cerca de 250 bytes) e uma frequência de chaveamento diferente.

3.11 Aquisição dos Códigos para controle via IR

Para o controle de televisores e condicionadores de ar é necessário saber o que é enviado pelo controle remoto dos equipamentos. Isso foi feito por engenharia reversa, com o auxílio de uma biblioteca para os televisores e receptores e com um código adaptado para os ares-condicionados.

3.11.1 Televisores e Receptores

Foi utilizada uma biblioteca para Arduino chamada IRremote[17] que amostra uma entrada digital onde é conectado um LED receptor de IR. A amostragem é feita a 38 kHz que é a frequência padrão de controles remotos e a biblioteca é capaz de comprimir os códigos amostrados em padrões. Existem 4 codificações reconhecidas pelo programa (SHIRRIFF, 2009):

- NEC
- SONY
- RC 5
- RC6

Para NEC é possível comprimir a amostragem de 38 kHz em 32 bits de mensagem sendo que quando é enviado o valor lógico 1 e o valor lógico 0 a mensagem fica conforme Figura 30.

[17] – Site com o projeto IRremote: https://github.com/shirriff/Arduino-IRremote

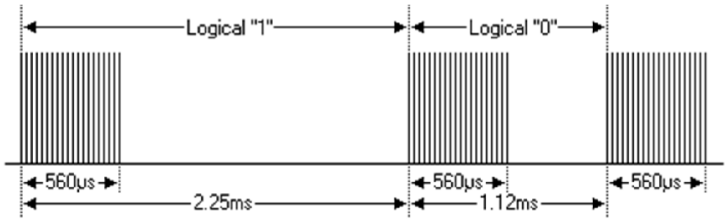

Figura 30 - Níveis lógicos do protocolo NEC.
Fonte: (SBPROJECT, 2013)

Ou seja, a cada 86 pulsos lidos pela entrada digital do Arduino é escrito um bit no comando NEC que possui 32bits. Para o envio a biblioteca simplesmente lê esses bits e reconstrói o trem de pulsos para ser enviado a uma saída digital (SBPROJECT, 2013).

Para os modelos SONY, RC5, RC6 e outros modelos não implementados a modulação é um pouco diferente porém o processo é semelhante, faz-se uma leitura a 38 kHz e existe um padrão de leituras que formam o 0 e o 1 lógico. A biblioteca comprime os pulsos para 0 e 1 lógicos e depois reconstrói o sinal. A Figura 31 mostra a ideia do sinal sendo enviado segundo uma modulação e depois no receptor um circuito de demodulação transforma esse sinal em níveis lógicos. Caso sejam perdidos alguns pulsos o circuito ainda assim consegue entender a mensagem.

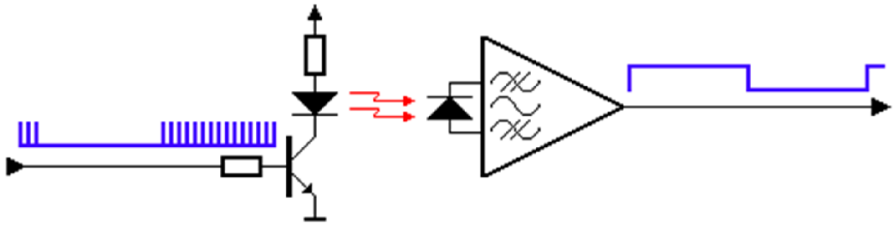

Figura 31 - Sinal enviado modulado e receptor demodulando o sinal.
Fonte: (SBPROJECT, 2011)

3.11.2 Ar-condicionado

Como as fabricantes de condicionadores de ar ainda não liberaram um protocolo padrão para seus controles, exceto a Carrier, o sistema tem que salvar todos os pulsos modulados para fazer o reenvio desses quando o código for solicitado.

Além disso, existem inúmeras combinações possíveis para serem montadas em um controle de ar-condicionado, já que a cada comando é enviada a temperatura, o modo de operação, a velocidade do ventilador e a presença de temporização para ligar ou desligar. Então optou-se por manter os códigos de temperatura e modo dentro do próprio controlador, assim evitando que mensagens muito longas transitassem pela rede sem fio. No momento que o usuário escolhe a temperatura o controlador automaticamente monta a mensagem e a envia via IR.

4 PROTÓTIPOS E RESULTADOS EXPERIMENTAIS

Nessa seção serão abordados os protótipos construídos, os custos envolvidos nesta construção e os resultados obtidos com base em testes feitos com estes protótipos. Os protótipos escolhidos são apenas dispositivos atuadores, ou seja, dependem de um comando do usuário e não fazem o controle de forma automática mas a rede de comunicação é bidirecional e aceita que seus dispositivos façam controle automático, como é o caso da proposta de controle de iluminação pela iluminancia do ambiente. Assim para a validação do trabalho foi montado um sistema de automação residencial contendo:

- Dois controladores para acionamento de dispositivos com controle infravermelho;
- Um controlador para controle de Iluminação;
- Uma central para interface do celular com o sistema de automação.
- Um Arduino Central para interface da Central Local com o sistema de automação.

Foram feitos testes relacionados a criptografia das mensagens RF e sua interpretação, teste de alcance dos módulos, ensaio de longa duração, consumo e levantamento de custos. A Figura 32 mostra um diagrama do sistema montado para a obtenção dos resultados experimentais.

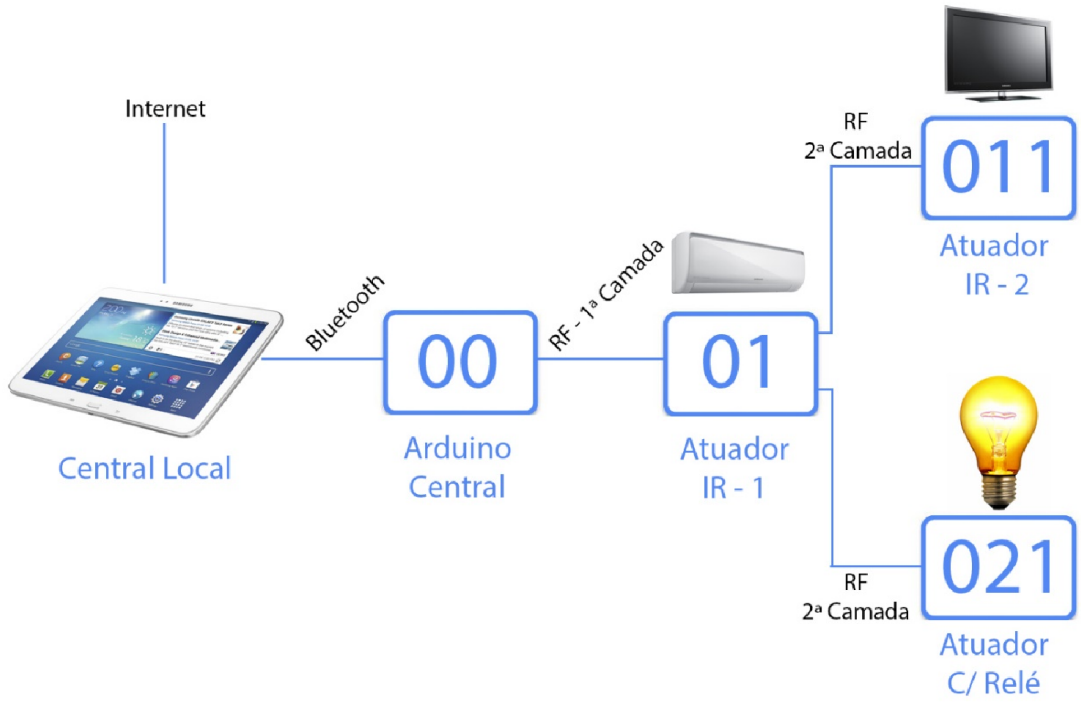

Figura 32 - Rede para Testes

4.1 Apresentação dos protótipos

Os protótipos de controladores foram montados dentro de caixas para que assim pudessem ser instalados para os testes de longa duração. Com os dispositivos montados dentro de caixas também foi possível avaliar o tamanho final destes. Nas próximas duas subseções estão os protótipos de controle de iluminação e dispositivos com controle infravermelho. Juntamente com imagens dos protótipos está um esquema de montagem/utilização para estes.

4.1.1 Arduino Central

O Arduino central por possuir um fluxo maior de dados, foi construído com um Arduino mega, ele também necessita de um módulo *bluetooth* que nenhum outro dispositivo da rede necessita sendo assim sua caixa é um pouco maior que as dos demais componentes. Uma imagem do Arduino Central pode ser vista na Figura 33. Este dispositivo possui apenas dois fios saindo da caixa que são utilizados para a alimentação do dispositivo.

Figura 33 - Arduino Central

4.1.2 Controlador de Iluminação

Este controlador como já descrito anteriormente é dotado de um relé para controlar um ponto de luz, sendo assim estão disponíveis para o exterior 5 fios. Um par de fios brancos conjugados que é utilizado para alimentação. Um fio vermelho que é o pino comum do relé, um fio branco que é o contato normalmente aberto do relé e um preto que é o contato normalmente

fechado do relé. A Figura 34 mostra o protótipo montado. O tamanho da caixa foi escolhido de forma a caber dentro de uma caixa de luz.

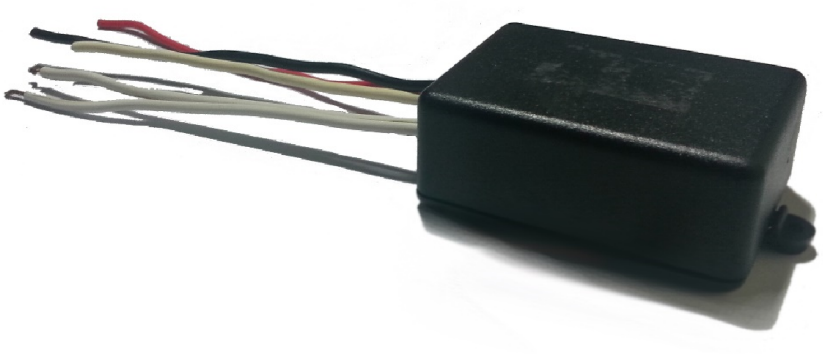

Figura 34 - Controlador para Controle de Iluminação

O dispositivo é alimentado com tensão alternada de 85~220V, e é possível utilizar os 3 contatos do relé formando uma chave hotel com um interruptor compatível, isso garante que o usuário sempre terá um controle manual da iluminação independentemente do funcionamento do circuito de automação. Os dois esquemas de ligação estão retratados nas figuras a seguir.

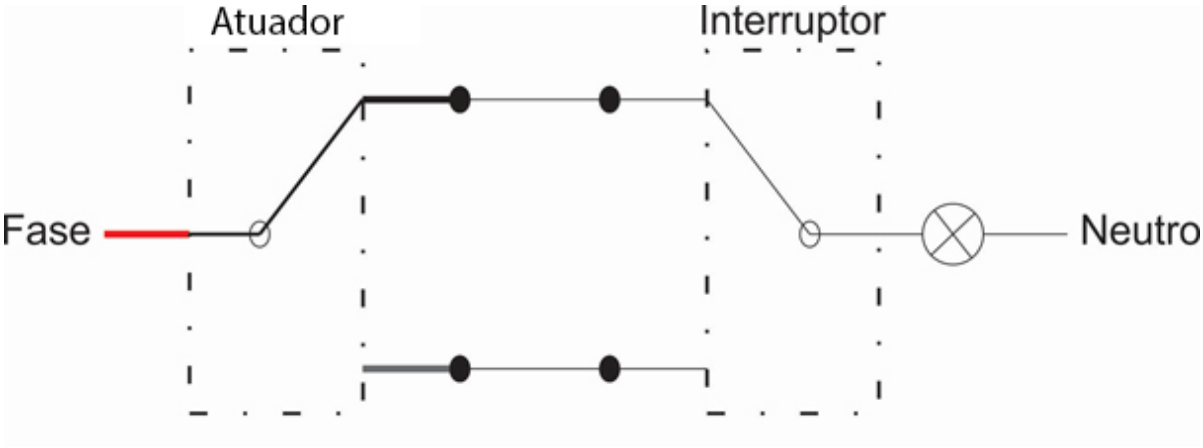

Figura 35 - Conexão com chave hotel

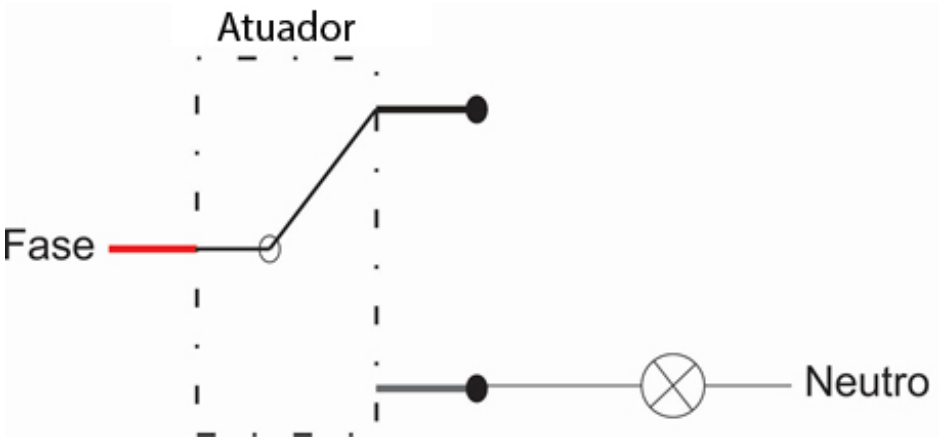

Figura 36 – Conexão apenas com o controlador

4.1.3 Controlador para controle via infravermelho

Este controlador, assim como o anteriormente apresentado, é alimentado pela rede com tensão alternada de 85~220V. Possui um LED infravermelho que precisa estar posicionado próximo, distância menor que 15 cm, ao receptor do aparelho a ser controlado. A imagem a seguir mostra o controlador. É possível notar na imagem o LED está posicionado em uma abertura da caixa.

Figura 37 - Controlador para controle via infravermelho

4.2 Estimativa de custos

Para comprovar a viabilidade econômica do trabalho é necessário que se faça uma análise de custos do projeto. Dada a indisponibilidade de alguns produtos no mercado nacional, foram feitas cotações dos produtos em fornecedores internacionais e então aos valores obtidos foram somados os impostos de importação, 60%, e de circulação de mercadorias ou serviços (ICMS), 18%.

Todos os produtos foram inicialmente cotados em dólar e para a conversão para reais foi utilizada a cotação oficial do dia de 06/11/2014 que foi de 2,55 reais por dólar. Os valores observados nas tabelas já estão convertidos para reais.

4.2.1 Custo do Arduino Central

Tabela 8 - Custo do Arduino Central

Descrição	Quantidade	Valor (R$)
Arduino Mega 2560	1	38,47
Módulo Bluetooth HC-06	1	15,31
Fonte de Alimentação (Flyback 220~85V – 5V)	1	6,78
Módulo RF (nrf24l01+)	1	10,00
Caixa de plástico	1	6,22
	Valor Total	76,78
	Valor com Impostos	149,81

4.2.2 Custo dos Controladores

Existem dois tipos de controladores no trabalho proposto, um para controle de iluminação e outro para dispositivos que possuam infravermelho, como televisões e aparelhos de ar condicionado.

Tabela 9 - Custo do Controlador IR

Descrição	Quantidade	Valor(R$)
Arduino Pro Mini	1	6,00
Módulo RF (nrf24l01+)	1	10,00
Fonte de Alimentação (Flyback 220~85V – 5V)	1	6,00
Resistor	2	0,20
Transistor	1	0,50
Led IR	1	0,30
Caixa de plástico	1	2,00
	Valor Total	25,00
	Valor com Impostos	48,78

Tabela 10 - Custo do Controlador com Relé

Descrição	Quantidade	Valor(R$)
Arduino Pro Mini	1	6,00
Módulo RF (nrf24l01+)	1	10,00
Fonte de Alimentação (Flyback 220~85V – 5V)	1	6,00
Modulo de Relé	1	3,50
Caixa de plástico	1	2,00
	Valor Total	25,50
	Valor com Impostos	49,75

4.2.3 Custo Total do Sistema

Para um simples sistemas de automação foi considerada a automação de um televisor, um ponto de iluminação e um ar-condicionado. Para esta configuração são necessários dois controladores de IR para controlar televisão e ar-condicionado e um com relé para controlar a iluminação, ainda é preciso de um *tablet* para Central Local. O custo do projeto ficou conforme a Tabela 11.

Tabela 11 - Custo total do sistema

Descrição	Quantidade	Valor(R$)
Central Local (*tablet*)	1	300,00
Arduino Central	1	149,71
Controlador IR	2	48,78
Controlador com Relé	1	49,75
	Valor Total	597,02

4.3 Bancada de Testes

A bancada de testes é formada por um *tablet* que funciona como Central Local, um celular para fazer o papel do dispositivo do usuário remoto, o Arduino Central responsável pelo repasse dos dados da Central Local para a rede RF, dois controladores com infravermelho e um com relé.

Como deseja-se ter acesso dos dados internos dos microcontroladores durante alguns testes, os dispositivos foram mantidos fora de suas caixas, permitindo dessa forma com que os pinos referentes a interface serial estivessem disponíveis.

No caso dos controladores que utilizam Arduinos Pro Mini é necessário ainda a conexão de um CI para fazer a conversão de UART para USB. O *setup* montado está mostrado na Figura 38.

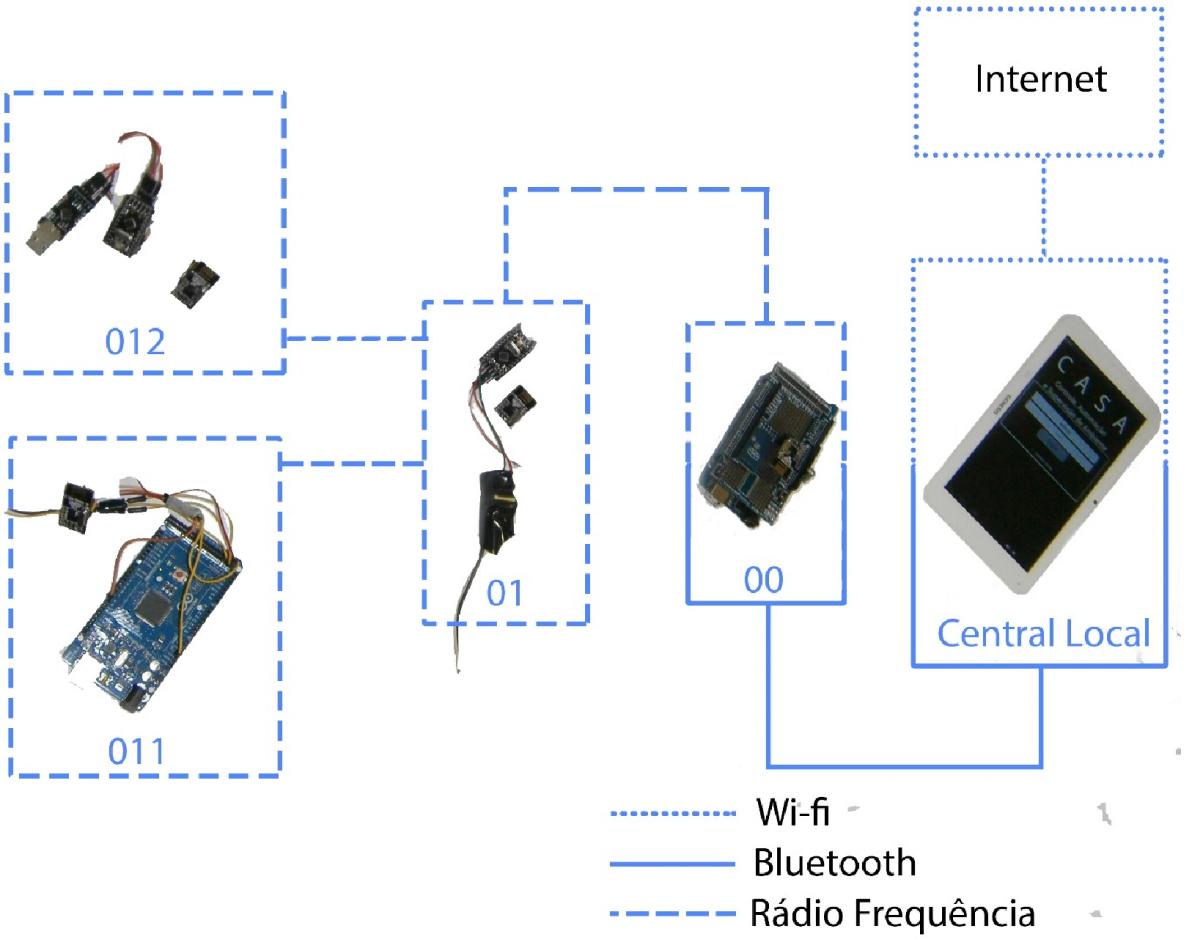

Figura 38 - Bancada de Testes

Os dispositivos durante os testes de comunicação foram alimentados por meio da própria serial do computador, dispensando o uso de uma fonte externa, porém as versões finais dos protótipos possuem alimentação por meio de fonte isolada.

Na bancada de testes ainda foi utilizado um Arduino MEGA como microcontrolador substituto ao Pro Mini, já que este não requer o uso de CI FTDI externo a sua placa para a conexão com o computador, assim simplificando a montagem, o Arduino utilizado para o controlador 011 da bancada de testes é um exemplo disso, ele está em destaque na Figura 39.

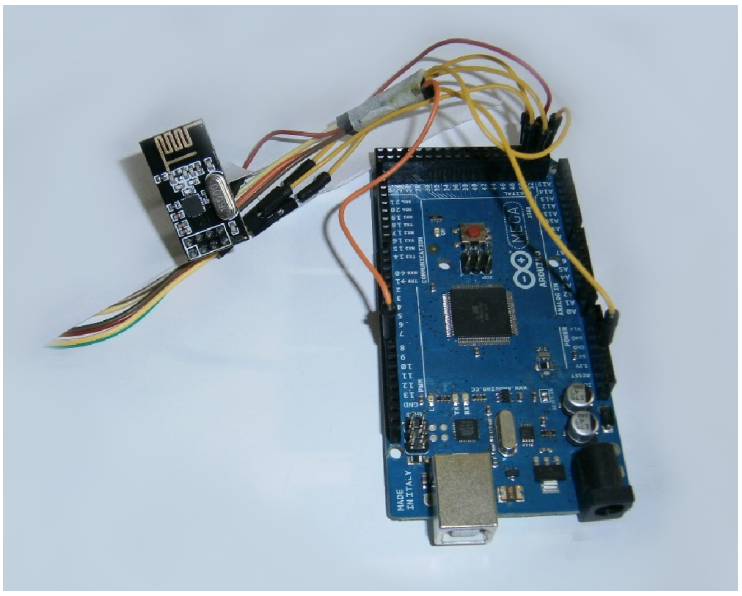

Figura 39 - Arduino Mega com módulo nrf24l01+

Mesmo com o uso de Arduinos Mega não se dispensou o uso de pelo menos um controlador utilizando Arduino Pro Mini, para a validação da topologia proposta. E que um dos Arduinos utiliza-se fonte de alimentação externa, estes dois exemplos são mostrados na Figura 40 e Figura 41 respectivamente.

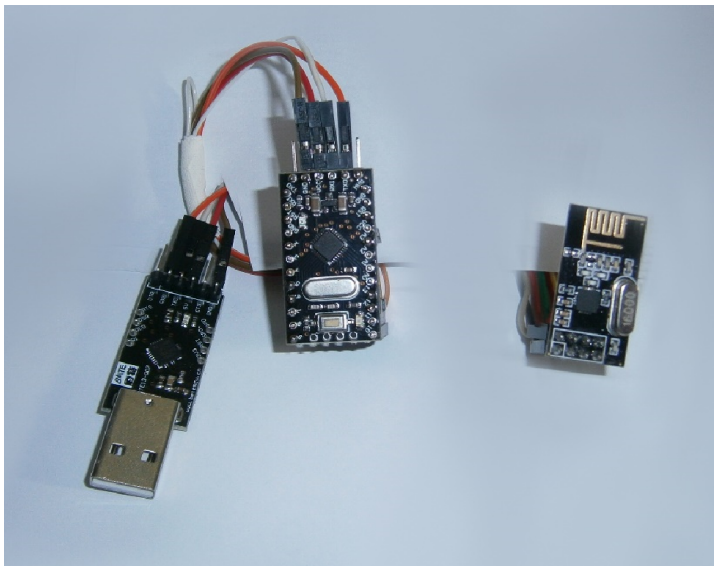

Figura 40 - Arduino Pro Mini com FTDI

Na Figura 40 é visto o dispositivo 012 da bancada de testes. É possível visualizar da esquerda para a direita o FTDI para conexão do Arduino com o computador, o Arduino Pro Mini e o módulo RF nrf24l01+.

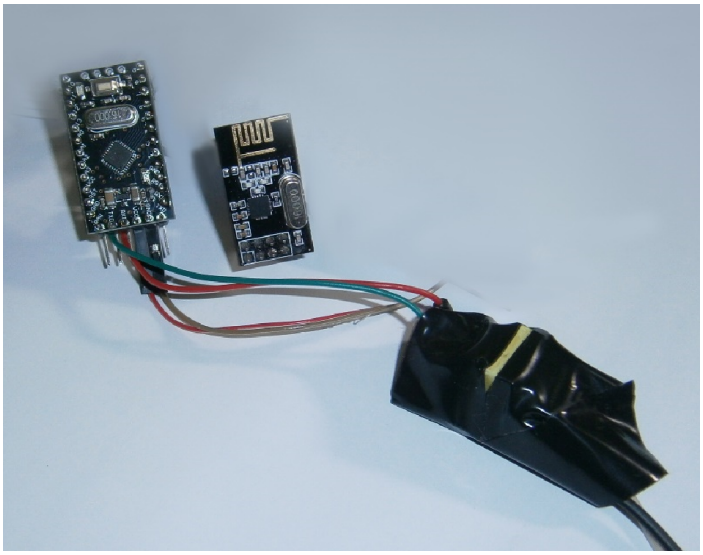

Figura 41 - Arduino Pro Mini com fonte

Na Figura 41 está mostrado da esquerda para a direita, um Arduino Pro Mini, o módulo RF e a fonte de alimentação, este dispositivo é o 01 da bancada de testes.

Na Figura 42 é possível ver o *tablet* que tem a função de Central local. Ele está rodando o aplicativo de supervisão desenvolvido durante o trabalho.

Figura 42 - Central Local

O dispositivo 00 que é o Arduino Central pode ser visto na Figura 43, quase completamente montado, apenas faltando a fonte de alimentação, já que durante os testes este permaneceu conectado a um computador.

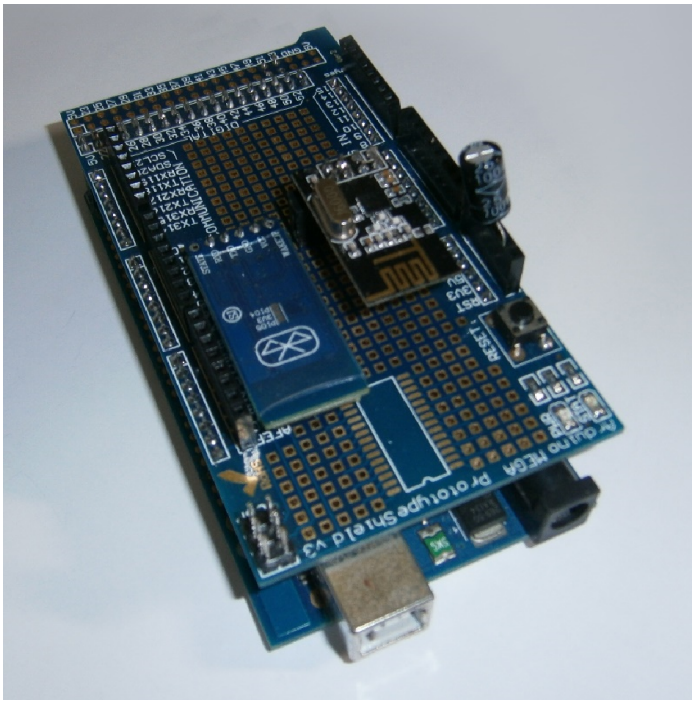

Figura 43 - Arduino Central

4.4 Teste de alcance

O teste de alcance pode ser dividido em duas partes, uma para o módulo *bluetooth* que comunica a Central Local com o Arduino Central e outro para os módulos RF. O teste da comunicação via *Bluetooth* é mais simples, utilizando uma velocidade de 9600 bps foi possível enviar mensagens a mais de 10 metros de distância com uma barreira (parede). Quando se aumentou a velocidade de envio para 19,2 kbps a distância de alcance se manteve quase inalterada.

Os módulos de rádio frequência se mostraram bem mais complexos para mensurar seu máximo alcance, sendo que o teste foi feito utilizando um método conhecido como PING-PONG onde um módulo envia uma mensagem e espera ouvir a resposta a esta mensagem do outro módulo. O alcance foi obtido considerando a distância onde o PONG parava de ser recebido de volta. Foi notada uma sensível variação de distância utilizando diferentes taxas de transmissão. Sendo obtida a tabela a seguir.

Tabela 12 - Alcance RF vs. Velocidade

Velocidade	Distância
250 kbps	30m
1 Mbps	15m
2 Mbps	10m

Porém os módulos também sofrem variação de distância pelo tamanho do pacote enviado. Quando o pacote é de 8 *bytes* o desempenho é conforme referido na Tabela 12, porém se aumentarmos o pacote de dados o alcance cai exponencialmente. No trabalho o pacote de dados é de 8 *bytes* e a 10 metros de distância utilizando uma velocidade de 1 Mbps a taxa de entrega das mensagens foi de 99%. O teste levou em consideração apenas uma parede entre os dois módulos e foram enviadas 1000 mensagens espaçadas de 2 segundos cada com apenas 10 mensagens sendo perdidas.

4.1 Método de teste dos códigos

Como não é possível saber de forma intrínseca o que está salvo em cada posição de memória sem o uso do JTAG do Arduino é necessário usar o artificio de enviar as posições de memórias (variáveis) que sejam interessantes no processo de testes do programa desenvolvido via código através de uma das seriais UART do microcontrolador.

Utilizando a própria IDE do microcontrolador (Arduino IDE) através da ferramenta "Serial Monitor" foram então coletados os dados enviados durantes os testes feitos no programa. Todos os testes mostrados a seguir possuem seus dados coletados através da técnica citada nesta seção.

4.2 Configuração dos módulos

Nesta subseção será mostrado o resultado da configuração dos módulos RF após a escolha dos parâmetros da rede. Entre os parâmetros configurados está a taxa de comunicação, o canal utilizado, modelo de módulo e tamanho da variável de checagem de erro. A Figura 44 mostra a configuração padrão utilizada para neste trabalho.

```
RF_CH             = 0x64
RF_SETUP           = 0x07
CONFIG            = 0x0f
DYNPD/FEATURE      = 0x00 0x00
Data Rate         = 1MBPS
Model             = nRF24L01+
CRC Length        = 16 bits
PA Power          = PA_HIGH
```

Figura 44 – Resultado da Configuração dos Módulos

É possível perceber na Figura 44 alguns dos principais parâmetros utilizados na configuração dos módulos, a seguir serão mostradas as opções de cada um desses parâmetros e o porquê da escolha destes como os padrões para este trabalho.

***RF_CH* – Canal de Rádio Frequência –** Especifica em que frequência o módulo vai operar, pois, este tipo de dispositivo pode operar entre 2,4 GHz e 2,525 GHz. A frequência do módulo é dada por 2,4 GHz mais o valor do canal escolhido que está em Mhz. No trabalho foi escolhido o canal 100, sendo assim os módulos estão operando em 2,5 GHz. Está escolha de frequência foi feita para que os módulos operem um canal mais limpo, tendo em vista que em 2,4 GHz existem redes *Wi-fi,* telefones sem fio, *bluetooth* e vários outros equipamentos, sendo assim 2,4 GHz é uma frequência muito poluída, diferentemente de 2,5 GHz.

***RF_SETUP* – Modo de Operação –** É responsável por informar se o módulo ficará todo o tempo em modo de recebimento/envio, ou se este terá momentos de hibernação esperando para ser acordado por uma mensagem via SPI.

***Data Rate* – Taxa de envio dos dados –** Define a velocidade com que os dados vão ser enviados. Como já foi mostrado na seção anterior que a velocidade de 1 Mbps possui confiabilidade e alcance adequados a uma aplicação residencial está foi a velocidade escolhida. O uso de um taxa de comunicação menor permite um alcance maior porem também gera mensagens mais lentas que ficam ocupando o canal por um período maior. Sendo assim como o trabalho não implementou um algoritmo de acesso ao meio foi utilizado o artificio de reduzir o tempo de ocupação do canal para evitar colisão de mensagens.

***CRC length* – Tamanho do CRC –** Tamanho da variável de checagem da integridade dos dados. Para um pacote de dados com 8 *bytes* ou mais é necessário um CRC de pelo menos 16 bits (2 *bytes*).

***PA Power* – Potência de amplificação –** Estes módulos possuem um ajuste na potência de envio das mensagens. É possível definir a potência como: Alta, Média, Baixa, Muito Baixa. No trabalho para garantir a integridade das mensagens foi utilizada a potência máxima do módulo.

4.3 Resultados da Troca de Mensagem

Como foi proposta no trabalho a rede de comunicação deveria ser criptografada com uma senha baseada no tempo, garantindo assim a segurança de rede. Desta forma nessa subseção serão comparadas algumas mensagens trocadas entre o Arduino Central e os Controladores.

Inicialmente a Central local e o Arduino central não estão sincronizados, portanto o Arduino central começa com uma data arbitrária logo que é ligado. Após intervalos definidos de tempo a Central Local sincroniza ou ressincroniza o Arduino Central. Este procedimento pode ser visualizado na Figura 45.

```
21:25:0 27/8/2013
Codigo gerado pelo TOPT: 411406
Baseado na estampa: 1377638700
Mensagem Bluetooth:0,30,23,54,22,14,11,14.
Sincronizando modulos nao criptografados
23:54:22 14/11/2014
Codigo gerado pelo TOPT: 966116
Baseado na estampa: 1416009262
```

Figura 45 - Sincronização do Arduino Central

Na Figura 45 é possível perceber, que como já havia sido mencionado, o Arduino Central é inicializado com uma data arbitrária sendo está 27/08/2013. Logo em seguida é recebida uma mensagem via *Bluetooth* proveniente da central local que informa a data correta. Logo após a sincronização o microcontrolador envia uma mensagem via rede RF para todos os outros controladores da rede afim de sincronizá-los. Essa mensagem é sem criptografia e é compreendida apenas pelos controladores ainda não sincronizados.

Os controladores enquanto não estão com o relógio ajustado ficam aguardando pelo sincronismo e portanto não executam nenhuma tarefa. Após o recebimento da mensagem de sincronismo não criptografada os controladores passam a operar normalmente, esperando algum comando valido via RF. A Figura 46 mostra o processo descrito anteriormente.

```
Aguardando Sincronizmo...
Sincronizado
Codigo gerado pelo TOPT: 966116
Baseado na estampa: 1416009262
23:54:22 14/11/2014
```

Figura 46 - Inicialização de um controlador

Com os processos de inicialização da central e do controlador já executados foram feitos então testes para validar a criptografia da mensagem. Para isso foi escolhida uma mensagem

destinada a ligar ou desligar um ponto de iluminação. Esta foi repetida em janelas de tempos diferentes e iguais para mostrar o algoritmo de criptografia em ação. A Tabela 13 mostra a troca de mensagens entre estes dispositivos.

Tabela 13 - Mensagem entre Central e Controlador

	Central			Controlador		
Tempo	Mensagem (Hex)	Senha (Dec)	Mensagem Cripto-grafada.	Mensagem Recebida (Hex)	Senha (Dec)	Mensagem Descripto-grafada (Hex)
14/11/2014 23:54:03	00 00 03 00 01 00 00 00	966116	38 7E 25 F8 A6 7D 79 F4	38 7E 25 F8 A6 7D 79 F4	966116	00 00 03 00 01 00 00 00
14/11/2014 23:54:25	00 00 03 00 01 00 00 00	966116	38 7E 25 F8 A6 7D 79 F4	38 7E 25 F8 A6 7D 79 F4	966116	00 00 03 00 01 00 00 00
14/11/2014 23:54:45	00 00 03 00 01 00 00 00	429740	3B 3E D6 FD 7E D0 4C 2E	3B 3E D6 FD 7E D0 4C 2E	572921	00 00 03 00 01 00 00 00

A mensagem recebida pelo Arduino central é criptografada baseada na senha variante no tempo e enviada ao controlador. É possível notar que a integridade da mensagem é mantida entre Arduino Central e controlador. Como a senha é baseada no tempo e os dispositivos estão sincronizados a mensagem é descriptografada e o comando é interpretado com precisão.

É possível notar que as mensagens 1 e 2 estão separadas de menos de 30 segundos sendo assim a senha é igual nas duas, já a terceira mensagem possui uma variação na senha. Essa janela de 30 segundos confere robustez ao algoritmo contra pequenas faltas de sincronismo entre os controladores e a central. A variação de senha mostrada entre as mensagens 2 e 3 prova que é possível proteger a comunicação contra mensagens replicadas garantindo que o sistema seja praticamente imune a invasões quando comparado a uma rede sem fio sem criptografia.

CONCLUSÃO

O projeto mostrou ser um sistema de baixo custo para a automação residencial, onde é possível fazer uma automação de uma residência de forma simples e com custos de instalação praticamente nulos já que toda a parte de transmissão de dados é sem fio e a alimentação dos dispositivos é feita utilizando diretamente a rede elétrica dispensando o uso de fontes auxiliares.

Porém o principal motivo para redução do custo na automação foi a utilização de uma IHM baseada em um dispositivo de massa que além de possuir baixo custo é uma tecnologia bastante acessível, este dispositivo é um *tablet* com Android. Em outros sistemas de automação seria necessária a compra de uma IHM dedicada o que acaba por representar um custo muito alto para a automação de uma casa.

Outra vantagem apresentada pelo sistema desenvolvido é o uso do celular para controlar a casa já que praticamente todas as pessoas estão sempre com ele ao alcance da mão. Desta forma o projeto torna-se prático, sem a necessidade de adicionar mais um dispositivo ao sistema. Outra característica é que o sistema proposto pode tanto suprir simples projetos de automação como apenas ligar e desligar luzes e também, atender a uma automação residencial completa com diversos outros dispositivos sendo controlados. A proposta de demonstração da rede com controle de iluminação, multimídia e temperatura é só um exemplo das possibilidades do projeto.

Utilizando a plataforma desenvolvida ao longo deste trabalho pode-se fazer um sistema mais complexo com novos controladores para comando de outros dispositivos e a geração de modos de cenas, simulação de presença e uma possível integração de sistemas de segurança e controle de acesso.

O trabalho também possibilita o acesso remoto a casa sem a necessidade que o usuário mantenha um servidor em seu domicilio, reduzindo não apenas o investimento inicial para a instalação, mas também os custos relacionados a manutenção deste.

O uso de roteamento em uma rede de comunicação também é uma importante contribuição do trabalho, se avaliarmos o custo dos módulos envolvidos nesta rede. Tecnologias como Zigbee e o Z-wave tem um custo de pelo menos o quatro vezes para uma mesma potência. Se formos considerar os kits de desenvolvimento ou módulos educacionais da tecnologia Zigbee esta diferença se torna muito maior.

Outra contribuição do trabalho é no que se refere ao uso de mensagens criptografadas garantindo maior privacidade e segurança ao usuário. Este tipo de criptografia não é muito

utilizada em automação residencial, estando disponível em poucos tipos de comunicação. O uso de uma criptografia simétrica (mesma chave para criptografar e descriptografar a mensagem) não garante segurança contra mensagens replicadas se não houver variação da mensagem, porém o uso da variação da chave ao longo do tempo, torna o sistema robusto contra este tipo de técnica. Assim o objetivo desejado com a codificação da mensagem foi alcançado.

Por fim o trabalho ofereceu uma rede modular de controladores que pode ser facilmente expandida com pequenas alterações nos programas. A topologia proposta possui um alcance razoável (40 m) mesmo com módulos de baixíssima potência (no máximo 1 mW) e tem mensagens seguras por meio de algoritmo de criptografia com chave variável no tempo.

REFERÊNCIAS

AIRSPAYCE PTY LTD. VirtualWire. **Air Spayce**, 2013. Disponivel em: <http://www.airspayce.com/mikem/arduino/VirtualWire/>. Acesso em: 12 mar. 2014.

BLUETOOTH SIG, INC. Adopted Specificaiton. **Bluetooth Special Interest Group**, 03 Dezembro 2013. Disponivel em: <https://www.bluetooth.org/en-us/specification/adopted-specifications>. Acesso em: 21 abr. 2014.

ELECTRON. & TELECOMMUN. RES. INST., DAEJEON. Remote-controllable and energy-saving room architecture based on ZigBee communication, v. 55 , n. 1, 2009.

FIPS. **FIPS 46-3**. Federal Information Processing Standards. [S.l.]. 1999.

GUANGZHOU HC INFORMATION TECHNOLOGY CO., LTD. **HC Serial Bluetooth Products - User Guide**. Xangai. 2010.

GUNGOR, V. C.; HANCKE, G. P. Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches. **Industrial Electronics, IEEE**, v. 56, n. 10, p. 4258-4265, Setembro 2009. ISSN 0278-0046.

INTERNATIONAL STANDARD. **Open Systems Interconection - Basic Reference Model: The Basic Model**. ISO/IEC. [S.l.], p. 68. 1994. (ISO/IEC 7498-1:1994(E)).

INTERNET ENGINEERING TASK FORCE (IETF). RFC3174. **US Secure Hash Algorithm 1 (SHA1)**, 2001. Disponivel em: <http://tools.ietf.org/html/rfc3174>. Acesso em: 26 mar. 2014.

INTERNET ENGINEERING TASK FORCE (IETF). RFC6238. **IETF Tools**, 2011. Disponivel em: <http://tools.ietf.org/html/rfc6238>. Acesso em: 01 abr. 2014.

ISO - INTERNATIONAL STANDAR ORGANIZATION. **ISO 8601**. Zurique. 2004.

KNIGHT, M. Wireless security - How safe is Z-wave? **Computing & Control Engineering Journal**, v. 17, n. 6, p. 18-23, Dezembro-Janeiro 2006. ISSN 0956-3385.

LEE, K. Y.; CHOI, J. W. **Remote-controlled home automation system via Bluetooth home network**. SICE 2003 Annual Conference. Fukui: IEEE Transactions. 2003.

LIAO, R.; WELLET, C.; EMMEL, W. Demystifying IEEE 802.11 for industrial wireless LANs. **Industrial Ethernet Book**, Março, n. 25, 2005.

MANIACBUG. **RF24Network for Wireless Sensor Networking**, 2012. Disponivel em: <http://maniacbug.wordpress.com/2012/03/30/rf24network/>. Acesso em: 2 Novembro 2014.

METCALFE, B. Metcalfe's Law after 40 Years of Ethernet. **Computer, IEEE**, v. 46, n. 12, p. 26-31, Dezembro 2013. ISSN 0018-9162.

NATIONAL INSTITUTE STANDARDS AND TECHNOLOGY. Computer Sercurity Division, 03 Outubro 1999. Disponivel em: <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>. Acesso em: 01 abr. 2014.

RODRIGUES, F.; CARDEIRA, C.; CALADO, J. M. F. **The impact of Wireless Sensors in Buildings Automation**. Ibersensor. Lisboa: [s.n.]. 2010.

SBPROJECT. SB Project, 2011. Disponivel em: <http://www.sbprojects.com/knowledge/ir>. Acesso em: 24 mar. 2014.

SBPROJECT. SB Project, 2013. Disponivel em: <http://www.sbprojects.com/knowledge/ir/nec.php>. Acesso em: 24 mar. 2014.

SHIRRIFF, K. Ken Shirriff's blog. **A Multi-Protocol Infrared Remote Library for the Arduino**, 2009. Disponivel em: <http://www.righto.com/2009/08/multi-protocol-infrared-remote-library.html>. Acesso em: 24 mar. 2014.

SILVA, T. D. D.; CAMPOS, A. L. P.; LIMA, F. S. D. SISTEMA DE SIMULAÇÃO DE PRESENÇA RESIDENCIAL, v. 26, n. 1, 2010.

SINTAY, B. UNIX TIMESTAMP. **UNIX TIMESTAMP**, 2002. Disponivel em: <http://www.unixtimestamp.com/>. Acesso em: 01 abr. 2014.

TEZA, V. R. **Alguns aspectos sobre a automação residencial**. Florianópolis. 2002.

VALENTIM, F. D. O.; MUNARO, C. J. **Integrating Technologies for Building a Wireless Home**. Induscon. Juiz de Fora, Brazil: [s.n.]. 2014.

WANG, Y.; P. LYNCH, J.; LAW, K. H. Validation of an Integrated Network System for Real-Time Wireless Monitoring of Civil Structures , 2005.

APÊNDICE A (COMUNICAÇÃO BLUETOOTH)

Bluetooth é uma especificação industrial para áreas de redes pessoais sem fio (Wireless personal area networks – PANs). As especificações do Bluetooth foram desenvolvidas e licenciadas pelo "Bluetooth Special Interest Group".

O bluetooth é conhecido por possuir várias camadas de protocolos sendo estes divididos em:

- Protocolos do núcleo;
- Protocolo de substituição de cabo;
- Protocolo de controle de Telefonia;
- Protocolos adotados;

Para o trabalho os protocolos interessantes estão no núcleo onde existem cinco camadas, sendo estas:

- Bluetooth Radio - especifica detalhes da interface com o ar, incluindo frequência, salteamento, esquema de modulação e potência de transmissão.

- Baseband - fala sobre estabelecimento de conexão com uma piconet, endereçamento, formato do pacote, temporização e controle de energia.

- Link Manager Protocol (LMP) - estabelece a configuração do link entre dispositivos bluetooth e gerenciamento de links em andamento, incluindo aspectos de segurança (ex. autenticação e encriptação), e controle e negociação do tamanho do pacote da banda base

- Logical Link Control and Adaptation Protocol (L2CAP) - adapta os protocolos da camada superior à camada de banda base, fornecendo tanto serviços sem conexão quanto serviços orientados à conexão.

- Service Discovery Protocol (SDP) - manipula informações do dispositivo, serviços e consultas para características de serviço entre dois ou mais dispositivos Bluetooth.

Com destaque a camada LMP que garante a segurança intrínseca dessa comunicação, dispensando o uso dos algoritmos de TOTP e DES utilizados na comunicação via RF.

Qualquer dispositivo pode realizar uma varredura para encontrar outros dispositivos disponíveis para conexão, e qualquer dispositivo pode ser configurado para responder ou não a essas requisições. O uso dos dispositivos, porém, requer pareamento (conhecido também como "emparelhamento") ou aceitação do proprietário.

Parear dispositivos é o ato de estabelecer uma comunicação segura "aprendendo" (por entrada do usuário) uma senha secreta. (passkey). O dispositivo que deseja se comunicar com

outro dispositivo deve informar uma senha que também deve ser digitada no outro dispositivo. Assim, depois de emparelhar, os dispositivos lembram os nomes amigáveis dos outros e conectam-se de forma transparente todas as vezes, assim como reconhecemos nossos amigos. Como o endereço Bluetooth é permanente, o pareamento é preservado, mesmo se o nome de algum dos dispositivos trocar. Pareamentos podem ser apagados (e assim ter as autorizações de conexão removidas) a qualquer momento. Alguns dispositivos podem se conectar apenas com um dispositivo por vez, e a conexão a esses dispositivos impede que eles possam receber requisições de outros ou que fiquem visíveis para outros aparelhos que estiverem realizando varredura.

Cada dispositivo é dotado de um número único de 48 bits que serve de identificação, no formato 00:00:00:00:00:00. Esses números são denominados "Endereço de Bluetooth" (Bluetooth Address) e são únicos e exclusivos para cada dispositivo fabricado, assim como o Endereço MAC das placas de rede. Os endereços geralmente não são mostrados, e no seu lugar aparecerá o nome corriqueiro (legível) do dispositivo, no caso do projeto o nome é sempre CASA. Esse nome aparecerá na lista de dispositivos disponíveis de qualquer aparelho que efetuar uma varredura.

Essas e outras informações ligadas ao protocolo bluetooth podem ser vistas em um documento disponibilizado pela Bluetooth Special Interest Group chamado de Adopted Bluetooth® Core Specifications (BLUETOOTH SIG, INC., 2013).

APÊNDICE B (CONFIGURAÇÃO DE MÓDULOS BLUETOOTH)

A configuração do módulo bluetooth é feita por meio de uma porta serial disponível no módulo, para a configuração são utilizados comandos AT, os comandos necessários para o projeto são os seguintes:

- AT+PE (para configurar a paridade para par)
- AT+PINxxxx (senha do módulo onde xxxx pode ser substituído pela senha)
- AT+NAMExxxx (nome do módulo onde xxxx pode ser substituído pelo nome que pode ser de até 20 caracteres)

Um simples exemplo de código para parametrização do módulo utilizando um Arduino é mostrado abaixo:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX
String command = ""; // Guarda a resposta do módulo
void setup()  {
        Serial.begin(115200);
        Serial.println("Digite os comandos AT!");
        mySerial.begin(9600);}
void loop()  {
        //Lê a saída se disponível
        if (mySerial.available()) {
                while(mySerial.available()) { // Continua lendo até o fim do pacote.
                        command += (char)mySerial.read();}
                Serial.println(command);
                command = "";  }
        if (Serial.available()){
                mySerial.write(Serial.read());}}
```

Outros comandos para configurar outras características estão no manual do usuário (GUANGZHOU HC INFORMATION TECHNOLOGY CO., LTD, 2010).

APÊNDICE C (MÓDULOS RF)

Os módulos nRF24L01+ possui as seguintes especificações:

- Rádio

2,4GHz ISM

126 canais RF (2,4 - 2,525GHz)

Tipo de Modulação: GFSK

Taxa de Dados sem fio: 250kbps, 1Mbps e 2Mbps.

Máxima taxa de envio: 1MHz sem sobre preposição de mensagem a 1Mbps

Máxima taxa de envio: 2MHz sem sobre preposição de mensagem a 2Mbps

- Transmissor

Potência de saída programada: 0 (1 mW), -6 (0,25 mW), -12 (62,5 μW) ou -18dBm (15 μW)

- Receptor

Sensibilidade da recepção:

-82dBm a 2Mbps

-85dBm a 1Mbps

-94dBm a 250kbps

Tamanho da mensagem dinâmico de 1 a 32 bytes

- Gestão de Energia

Regulador de tensão integrado

Alimentação em 1.9 a 3.6V

26μA em Standby

900nA desligado

- Interface Local

SPI de 4 pinos

Taxa de Dados: Max 10Mbps

3 FIFOs independentes de 32 bytes para TX e RX

Entradas tolerantes a 5V

APÊNDICE D (TOTP)

O tipo de geração de senha utilizada foi uma senha única gerada com base em uma chave comum a rede e um fator baseado no tempo (INTERNET ENGINEERING TASK FORCE (IETF), 2011), sendo assim definimos a função geradora da senha como sendo uma:

$$TOTP(k, t)$$

Onde k é uma chave de 10 bytes e t é um valor baseado na data, esse valor é conhecido como estampa de tempo. O padrão de estampa de tempo utilizado é o da ISO 8601 que utiliza o padrão UNIX.TIMESTAMP. O número gerado por essa estampa de tempo é a quantidade de segundos desde primeiro de janeiro de 1970 em UTC (ISO - INTERNATIONAL STANDAR ORGANIZATION, 2004). Como a variável conta até valores de 32bits temos que a ultima senha não repetida gerada pelo sistema vai ser em 19 de janeiro de 2038, então o sistema sofrerá um overflow da variável e as senhas voltam a se repetir (SINTAY, 2002).

Com base nessa estampa de tempo e na chave, roda-se um algoritmo de criptografia chamado de SHA-1, onde é gerado então o código de 6 bytes. (INTERNET ENGINEERING TASK FORCE (IETF), 2001)

O SHA-1 é um algoritmo de HASH que baseado em uma variável de entrada que pode ter diferentes tamanhos e uma senha de tamanho fixo que gera uma saída de tamanho sempre fixo. No projeto, por exemplo, temos um algoritmo que recebe uma senha de 10 bytes fixa e um dado variável de 1-4 bytes(timestamp), nesse caso então é gerado um código sempre de 6 bytes.

APÊNDICE E (DES)

O algoritmo criptografa blocos de 64bits, dado que as mensagens transmitidas no trabalho tem esse mesmo tamanho, não há necessidade de quebrar as mensagens em mais blocos. O algoritmo baseia-se em dois blocos de 32 bits que são gerados quebrando a mensagem em parte alta e baixa. Existe a aplicação de uma das partes (baixa) em uma função chamada de Feistel (F) e então a XOR (⊕) bit-a-bit com a parte alta da mensagem. A parte baixa então permutasse com a alta, e a parte gerada após a XOR assume o lugar da parte baixa. O procedimento é repetido por 16 vezes até se ter a FP que é a mensagem de saída criptografada. O procedimento oposto pode ser feito para descriptografia da mensagem. O procedimento citado anteriormente pode ser observado na Figura 47.

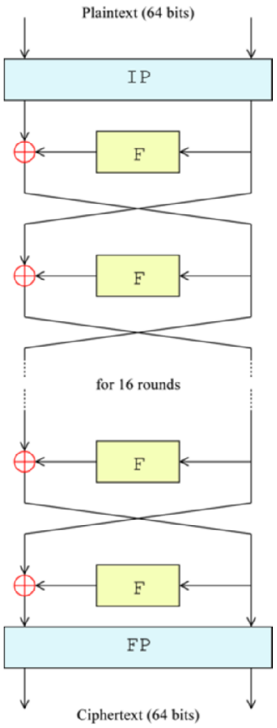

Figura 47 – Algoritmo DES.

A função Feistel(F) é onde aplicasse a chave de criptografia, no caso de sistema do trabalho essa chave é variável de acordo com o algoritmo TOTP. A F pode ser definida por 4 estágios sendo:

Expansão – onde a mensagem é expandida de 32 bits para 48 bits, para ficar compatível com o tamanho da senha que é de 6 bytes como já explicado no Apêndice D. Essa expansão se da por permutação de expansão e é representada por E no diagrama de blocos. A expansão por permutação é feita com cada meio byte de entrada sendo montado agora com uma copia de seus 4 bits mais um bit antecessor e um sucessor destes 4 bits. Como temos 4 bytes ou seja 32 bits e

a cada 4 bits são adicionados outros 2 bits terminamos com 48bits valor compatível com os 6 bytes da senha.

Mistura com a Senha – É feita uma XOR bit-a-bit com a senha portanto gerando outros 48bits agora misturados com a senha.

Substituição – é necessário retornar para uma variável de 32bits para isso a cada 6 bits devem ser substituídos por uma sequência de 4 bits, isso é feito por meio de uma caixa de substituição Sbox 6x4. Esse tipo caixa possui uma tabela padrão que correlaciona entrada(6bits) com saída(4bits).

Permutação – Antes de agrupar as saídas dos blocos Sbox eles passam por um processo de permutação entre seus bits.

O diagrama que descreve a função F está na Figura 48.

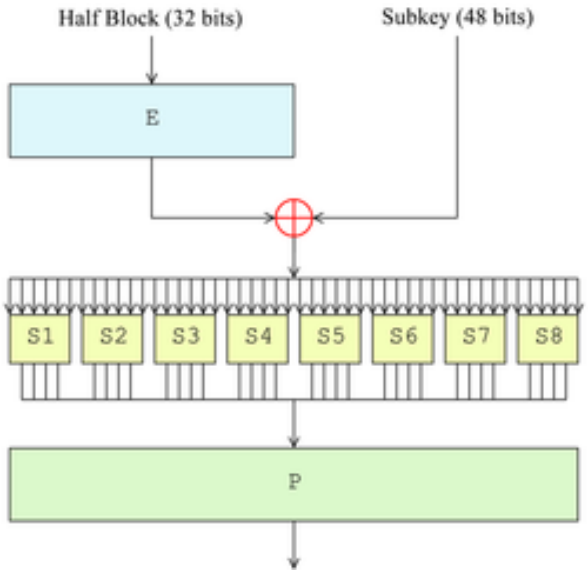

Figura 48 - Função de Festel

Estas e outras informações sobre o DES podem ser encontradas no (NATIONAL INSTITUTE STANDARDS AND TECHNOLOGY, 1999).