

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

BRUNA ROBERTA SEEWALD DA SILVA

**Sistema de automação residencial de baixo
custo para redes sem fio**

Trabalho de Graduação.

Prof. Dr. Sérgio Luis Cechin
Orientador

Porto Alegre, julho de 2014.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do ECP: Prof. Marcelo Götz

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

Agradecimentos

Gostaria de agradecer a Deus pela proteção durante todos esses anos, por me ajudar a encontrar a resposta quando muitas perguntas pareciam não ter solução e por me ajudar todos os dias a vencer os obstáculos que a vida nos traz.

Sou profundamente grata aos meus pais, que nunca mediram esforços para oferecer sempre o melhor a mim e ao meu irmão. O apoio, a confiança e principalmente o amor incondicional de vocês são imprescindíveis na minha caminhada pela vida. Obrigada por tudo.

Ao meu irmão, que é um homem gentil e carinhoso. Teu companherismo e compreensão são especiais demais para mim.

A minha avó, que sempre me apoiou. Você é um exemplo de vida, agradeço profundamente por tudo que fizestes por mim e dedico esta conquista para ti.

Gostaria também de agradecer a Ana Cláudia, Ana Clara, Helena e Natacha. Vocês são como uma segunda família para mim, sempre me apoiando desde o momento em que nos conhecemos. A amizade de vocês é extremamente importante para mim e por isso o meu muito obrigado.

A Luiza, que é um anjo em minha vida. Obrigada por tudo que tu fazes por mim e pela incrível vontade de sempre estar disposta a me ajudar. Tuas palavras e conselhos são essenciais na minha vida.

Por fim, agradeço a UFRGS e aos meus mestres, pela educação de qualidade e pela dedicação durante todos os meus anos na universidade.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS.....	6
LISTA DE FIGURAS.....	8
LISTA DE TABELAS.....	9
RESUMO.....	10
ABSTRACT.....	11
1 INTRODUÇÃO	12
2 PRINCIPAIS CONCEITOS RELACIONADOS.....	13
2.1 O padrão Wi-Fi.....	13
2.1.1 Camada Física.....	13
2.1.2 Camada de Enlace.....	14
2.1.3 As principais revisões da especificação IEEE 802.11	15
2.1.3.1 IEEE 802.11a.....	15
2.1.3.2 IEEE 802.11b.....	15
2.1.3.3 IEEE 802.11g.....	15
2.1.3.4 IEEE 802.11n.....	15
2.2 O padrão ZigBee.....	16
2.2.1 Camada Física.....	16
2.2.2 Camada de Enlace.....	17
2.2.3 Camada de Rede	17
2.2.4 Camada de Aplicação	17
2.2.5 Tipos de dispositivos ZigBee.....	18
2.2.6 Topologias de Rede	18
2.3 O módulo XBee	19
3 IMPLEMENTAÇÃO DO PROJETO	22
3.1 Arquitetura do Sistema.....	22
3.1.1 Arquitetura de Hardware	23
3.1.2 Arquitetura de Software.....	23
3.2 Circuitos e sensores	26
3.2.1 Controlador	26

3.2.2	Sensores	27
3.2.2.1	<i>Movimento</i>	27
3.2.2.2	<i>Gás</i>	27
3.2.2.3	<i>Luminosidade</i>	28
3.2.2.4	<i>Temperatura</i>	29
3.3	O Sistema de Automação	30
3.3.1	Aplicativo Android	30
3.3.1.1	<i>Smart House Control</i>	31
3.3.2	Módulo Controlador	32
3.3.3	Conectividade com a Nuvem	34
3.3.4	Configuração dos módulos XBee	36
3.4	Custo total do projeto.....	37
4	CONCLUSÃO.....	38
	REFERÊNCIAS.....	39
	ANEXO A CÓDIGO FONTE	40
	ANEXO B TRABALHO DE GRADUAÇÃO 1	42

LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledgement
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
API	Application Programming Interface
APS	Application Support Sublayer
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DCF	Distributed Coordination Function
DIO	Direct Input Output
DSSS	Direct Sequence Spread Spectrum
FCC	Federal Communications Commission
FDM	Frequency Division Multiplexing
GTS	Guaranteed Time Slot
GND	Ground
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial Scientific Medical
JSON	Javascript Object Notation
MAC	Medium Access Control
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open Systems Interconnect
PAN	Personal Area Network
PCF	Point Coordination Function
RTS	Request to Send
REST	Representation State Transfer
USB	Universal Serial Bus
VDD	Voltage Drain Drain

WLAN	Wireless Local Area Network
XML	Extensible Markup Language

LISTA DE FIGURAS

Figura 2.1: Rede com infraestrutura e rede AdHoc.....	13
Figura 2.2: Representação dos canais na faixa de 2,4 GHz.....	14
Figura 2.3: Pilha ZigBee.....	16
Figura 2.4: Topologia em estrela.....	18
Figura 2.5: Topologia em árvore.....	19
Figura 2.6: Topologia em malha.....	19
Figura 2.7: XBee Series 2.....	19
Figura 2.8: Estrutura do quadro no modo API.....	20
Figura 2.9: Exemplo de um quadro API.....	20
Figura 3.1: Arquitetura do sistema de automação residencial.....	22
Figura 3.2: Diagrama que mostra as conexões em nível de hardware.....	23
Figura 3.3: Arquitetura do aplicativo Android.....	24
Figura 3.4: Arquitetura do controlador.....	25
Figura 3.5: Circuito Controlador.....	26
Figura 3.6: Circuito com sensor de movimento.....	27
Figura 3.7: Divisor de tensão.....	28
Figura 3.8: Desenho do circuito com sensor de gás.....	28
Figura 3.9: Desenho do circuito com sensor de luz.....	29
Figura 3.10: Desenho do circuito com sensor de temperatura.....	29
Figura 3.11: Circuito único com os sensores de gás, luminosidade e temperatura.....	30
Figura 3.12: Tela de entrada e tela secundária.....	31
Figura 3.13: Trecho do código de acionamento do sensor de movimento.....	32
Figura 3.14: Sensores cadastrados na plataforma Carriots.....	34
Figura 3.15: Lista com os dados enviados para a Nuvem.....	35
Figura 3.16: Listeners que enviam mensagens para alertar o usuário.....	35
Figura 3.17: Listener para enviar email.....	35
Figura 3.18: Email alertando o usuário sobre o alarme ativado.....	36
Figura 3.19: Tela do software X-CTU.....	36

LISTA DE TABELAS

Tabela 2.1: As principais revisões e suas características.....	15
Tabela 2.2: Faixas de frequência do padrão IEEE 802.15.4.....	16

RESUMO

A automação residencial existe há muitos anos no mercado, no entanto ainda é considerada um artigo de luxo pela maioria dos consumidores. A falta de padronização dos equipamentos é um dos graves problemas nesta área e afeta diretamente o custo de implementação dos projetos. Este trabalho busca apresentar uma solução de automação residencial baseada em redes sem fio e que seja acessível em termos de custo.

O sistema é composto por sensores (movimento, gás, luminosidade e temperatura) controlados por meio de um aplicativo móvel para Android e utiliza os protocolos Wi-Fi e ZigBee para a transmissão de dados.

Para o projeto foram desenvolvidos circuitos com módulos XBee Series 2, o aplicativo móvel e o software do controlador (responsável pelo gerenciamento das informações no sistema de automação residencial). O projeto final permite que o sistema seja ampliado, oferecendo mais opções além das abordadas neste trabalho.

Palavras-Chave: automação residencial, redes sem fio, sensores, Android.

Low-cost home automation system for wireless networks

ABSTRACT

Home automation is still considered a luxury by most consumers, even though it exists for many years on the market. The lack of standardization is one of the most serious problems in this area and directly affects the cost of the projects. This study aims to present a solution for home automation which is based on wireless networks and accessible in terms of cost.

The system consists of sensors – motion, gas, light and temperature – controlled by a mobile app for Android. Wi-Fi and ZigBee protocols are used for data transmission.

For the project, circuits with XBee Series 2, the mobile application and the controller software – responsible for managing information in the home automation system – were developed. The final project allows the system to be expanded, offering more options beyond those discussed in this paper.

Keywords: Home automation, wireless networks, sensors, Android.

1 INTRODUÇÃO

A automação residencial cresceu muito nas últimas décadas e atualmente ocupa uma posição de destaque no mercado mundial. Um dos principais objetivos é proporcionar conforto para os moradores, auxiliando nas tarefas diárias e possibilitando a redução do consumo energético.

Impulsionadas por este crescimento, novas tecnologias se desenvolveram, com ênfase na área de redes sem fio. Protocolos como o ZigBee, Z-Wave e Bluetooth Low Energy competem pela preferência dos usuários no mercado pois oferecem maior flexibilidade sem a necessidade de grandes reformas nas residências. Além disso, os custos dos equipamentos reduziram significativamente, aumentando o interesse dos consumidores e tornando-a, conseqüentemente, mais acessível e popular.

Apesar deste crescimento a automação residencial possui um sério entrave relacionado à falta de padronização do protocolo utilizado. Até o momento não encontrou-se uma solução que seja ótima e atenda globalmente às necessidades do mercado, por isso muitos fabricantes preferem investir em soluções proprietárias para comercializarem em conjunto com os seus produtos, fator, inclusive, que faz com que o valor agregado do produto aumente muito.

Assim, a principal motivação deste trabalho é promover a automação residencial a partir de tecnologias existentes e de baixo custo. Para isso estudou-se uma variedade de protocolos - entre abertos e proprietários - e após análise das características de cada um escolheu-se o Wi-Fi e o ZigBee para integrarem o sistema de automação residencial proposto.

Entre os protocolos, escolheu-se o Wi-Fi devido à sua enorme disponibilidade nos dispositivos móveis como celulares e tablets e por ser uma das principais formas de conexão com a internet, presente na maioria das residências e estabelecimentos comerciais. Escolheu-se também o ZigBee pois ele oferece melhor eficiência energética, fator de extrema importância para redes que possuem sensores e para circuitos que são alimentados por baterias.

O sistema é composto por alguns sensores controlados por um aplicativo móvel para Android. Devido ao enfoque no baixo custo da automatização, foram desenvolvidos circuitos com componentes de preço acessível, além da utilização de protocolos abertos.

Um dos principais módulos deste trabalho é o Controlador, responsável pelo gerenciamento das informações no sistema de automação residencial.

2 PRINCIPAIS CONCEITOS RELACIONADOS

2.1 O padrão Wi-Fi

O padrão Wi-Fi foi estabelecido pela Wi-Fi Alliance e segue a norma IEEE 802.11. Esta norma especifica as camadas física e de enlace para a implementação de uma rede local sem fio (WLAN) nas frequências de banda de 2.4 GHz, 5 GHz e 60 GHz.

Neste padrão dois tipos de arquiteturas são possíveis: infraestruturada ou AdHoc. Na primeira os dispositivos estão conectados a pontos de acesso que são responsáveis pela comunicação e regulamentação da rede. Na segunda, os dispositivos podem operar como pontos de acesso, estabelecendo a comunicação entre eles. A figura 2.1 apresenta estas duas arquiteturas.

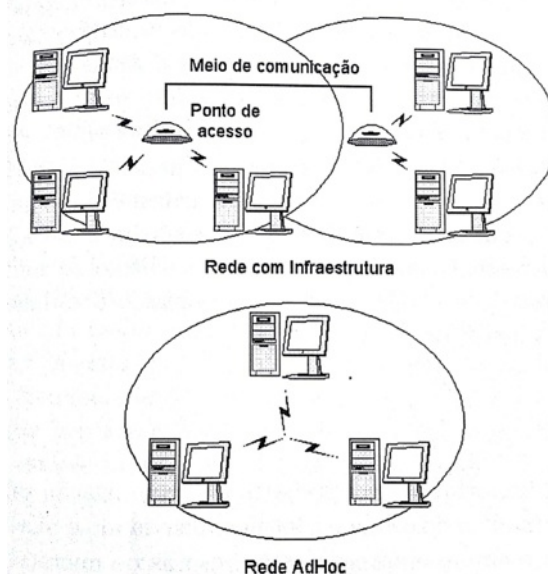


Figura 2.1: Rede com infraestrutura e rede AdHoc (BOLZANI, 2004).

2.1.1 Camada Física

Conforme o modelo OSI, esta camada é responsável pela transmissão bruta dos dados por meio de um canal de comunicação. Para isso, alguns parâmetros são definidos como, por exemplo, a modulação do sinal e a frequência da banda utilizada.

A primeira faixa - entre 2,400 e 2,500 GHz - abrange 14 canais de operação, sendo estes espaçados a cada 5 MHz, com exceção do último canal. Assim, o primeiro canal é centrado na frequência de 2,412 GHz e o último em 2,484 GHz. Esta faixa também é

conhecida como ISM (*Industrial - Scientific - Medical*) e por ser aberta e livre, pode sofrer interferências devido à sua utilização por outros padrões de rede e por eletrodomésticos. A figura 2.2 mostra a representação dos canais na faixa de 2,4GHz.

A segunda faixa, de 5 GHz, possui três divisões: 5,150 - 5,250 GHz (indoor), 5,250 - 5,350 (in/outdoor) e 5,725 - 5,825 GHz (outdoor point-to-point).

Apesar dessas definições, cada região possui um órgão responsável pela utilização destas frequências. Assim, algumas regras e a distribuição de canais podem variar entre os países. Nas Américas, por exemplo, o FCC (*Federal Communications Commission*) é o responsável pela regulamentação.

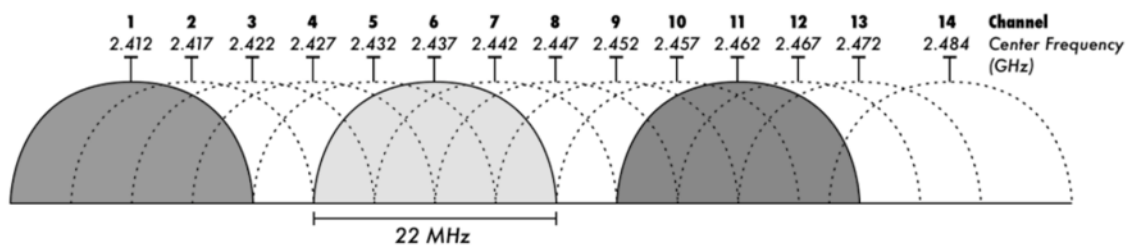


Figura 2.2: Representação dos canais na faixa de 2,4 GHz (CARISSIMI, 2011).

As duas técnicas de modulação espectral utilizadas são o DSSS (Direct Sequence Spread-Spectrum) e o OFDM (Orthogonal frequency-division multiplexing).

O DSSS foi utilizado na revisão IEEE 802.11b e possui três características muito importantes segundo (ROCHOL, 2011): a primeira consiste no fato de que a banda pode ser ocupada simultaneamente por diversos usuários, pois cada sinal é modulado conforme um código PN único e são ortogonais entre si. A segunda refere-se à potência de cada sinal, que pode ser abaixo da potência do ruído ou da interferência mútua, fato que é explicado pelo teorema de Shannon/Hartley sobre a capacidade máxima de um canal que apresenta uma determinada relação sinal/ruído. Já terceira atenta para o fato de que os sinais são muito robustos em relação às interferências, por isso o processo de recuperação filtra esses sinais que se confundem com o ruído de fundo.

O OFDM está presente nas atuais versões da norma IEEE 802.11 e, segundo (ROCHOL, 2011), é uma técnica de multiplexação FDM (Frequency Division Multiplex) altamente eficiente de múltiplas portadoras no domínio frequência, as quais podem ser partilhadas por múltiplos usuários. O acesso destes usuários a um mesmo canal é possível por meio da alocação dinâmica de conjuntos de subportadoras por usuário. Dependendo das exigências, é possível utilizar mais ou menos banda, alocando assim as respectivas subportadoras. Uma característica importante é a sua capacidade de ajustar automaticamente a taxa de modulação das subportadoras em função da relação sinal/ruído. Quanto maior a relação, maior a vazão de dados. Em contrapartida, quanto menor a relação, mais robusta deve ser a modulação e menor é a vazão de dados.

2.1.2 Camada de Enlace

Nesta camada é feito o controle de acesso ao meio por meio de dois modos de operação que trabalham em conjunto.

O primeiro é o DCF (Distributed Coordination Function), baseado no protocolo CSMA. Este modo é obrigatório para ambas as arquiteturas, infraestruturada e AdHoc, e é fortemente baseado em temporização.

O CSMA/CA é um método de transmissão com prevenção de colisões que utiliza um valor aleatório chamado de *backoff*. Caso o meio esteja ocioso, o valor de *backoff* é decrementado e quando este chegar a zero, o quadro é transmitido e espera-se pelo ACK. Caso o meio esteja ocupado, o valor é mantido. Também pode-se utilizar o conceito de alocação de um canal virtual com quadros de controle conhecidos por RTS e CTS.

O segundo modo de operação, chamado PCF (Point Coordination Function), é baseado em *polling*, ou seja, verifica periodicamente o canal. Assim, ao identificar que uma estação possui dados para transmitir, ela ganha acesso exclusivo ao meio. Ele é construído acima do DCF na subcamada MAC.

2.1.3 As principais revisões da especificação IEEE 802.11

Com o passar dos anos novas revisões aos padrões existentes foram feitas. As principais alterações ocorreram na frequência utilizada, na largura de banda, no tipo de modulação e, conseqüentemente, na taxa de transmissão.

2.1.3.1 IEEE 802.11a

Esta revisão foi realizada em paralelo com a norma IEEE 802.11b sendo, no entanto, aceita posteriormente. Entre as suas vantagens está o fato de operar na faixa dos 5 GHz e permitindo a transferência de taxas mais elevadas, porém seu alcance é menor.

2.1.3.2 IEEE 802.11b

Lançada em 1999, rapidamente tornou-se o padrão Wi-Fi. Apesar de as taxas de transmissão serem menores, o baixo custo dos componentes que compõe os equipamentos contribuiu para a sua rápida aceitação no mercado. Um dos pontos negativos desta norma é que ela opera na faixa de frequência de 2.4 GHz, a mesma que outros equipamentos eletrônicos e que pode ocasionar interferências entre os sinais.

2.1.3.3 IEEE 802.11g

Esta revisão é retrocompatível com a norma IEEE 802.11b porém possui modulação OFDM com taxas brutas de transmissão em torno de 54 Mbps. Ainda hoje é largamente comercializada em equipamentos com padrão Wi-Fi.

2.1.3.4 IEEE 802.11n

Esta norma contou com significativas alterações para suportar altas taxas de transmissão. Assim, foi introduzida a possibilidade de utilizar-se canais de 40 MHz, duplicando a taxa de transmissão por canal. Além disso, as entradas e saídas dos rádios, anteriormente alternadas e funcionando apenas uma por vez foram modificadas. Esta modulação acarretou em maior eficiência e permitiu o uso simultâneo de múltiplas antenas para transmissão/recepção dos sinais.

Tabela 2.1: As principais revisões e suas características

Padrão	Ano	Mbps (máx)	Mbps(típico)	Banda (GHz)	Modulação
802.11a	1999	54	23	5	OFDM
802.11b	1999	11	4.3	2.4	DSSS
802.11g	2004	54	19	2.4	OFDM
802.11n	2008	600	74	2.4/5	OFDM

Fonte: CARISSIMI, 2011.

2.2 O padrão ZigBee

O ZigBee foi desenvolvido pela ZigBee Alliance para comunicação em redes sem fio. Este padrão, apesar de aberto, requer uma assinatura anual para os produtos serem comercializados e certificados dentro dos padrões da ZigBee Alliance.

Seus principais propósitos são fornecer baixa latência e baixo consumo para equipamentos que não necessitam de altas taxas de transferência e que geralmente utilizam baterias. Este padrão segue a norma IEEE 802.15.4, que define as especificações para a camada física e de controle de acesso ao meio para redes pessoais com baixa taxa de transmissão.

A ZigBee Alliance ainda estende a pilha adicionando as camadas de rede e aplicação e, conforme (KINNEY, 2003), a simplicidade do protocolo faz com que ele seja muito menor que a pilha do Bluetooth. A figura 2.3 apresenta a pilha ZigBee.

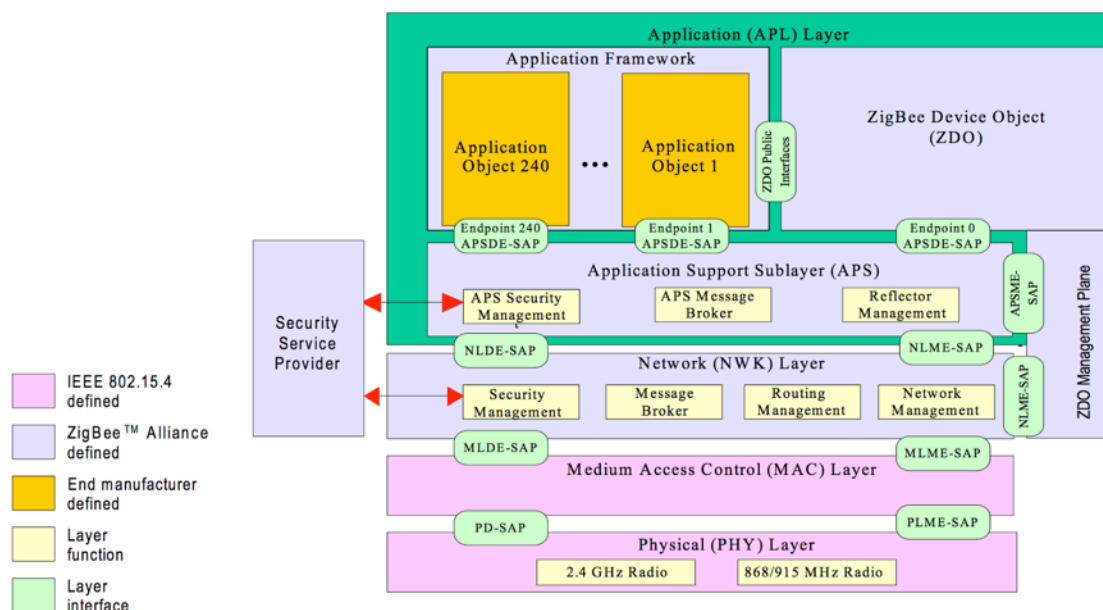


Figura 2.3: Pilha ZigBee (ASHTON, 2009).

2.2.1 Camada Física

Neste padrão são definidas três faixas de frequência que variam conforme a região. Na Europa apenas o canal 0 na frequência 868.3 MHz é utilizado. Nos Estados Unidos, por sua vez, utiliza-se a faixa que abrange de 902 MHz até 928MHz, sendo os canais espaçados a cada 2 MHz. No restante do mundo a frequência de 2.4 GHz é usada, com 16 canais espaçados a cada 5 MHz.

O método de espelhamento espectral é chamado de DSSS, variando a técnica de modulação para cada região.

Tabela 2.2: Faixas de frequência do padrão IEEE 802.15.4

Camada Física	Frequência	Canais	Taxa de bit (Kbits/s)	Modulação
868/915 MHz	868.3 MHz	0	20	BPSK
	902-928 MHz	1-10	40	BPSK
2.4 GHz	2.4-2.4835 GHz	11-26	250	O-QPSK

2.2.2 Camada de Enlace

Esta camada é responsável pelo gerenciamento do acesso ao meio, pela transmissão de *beacon frames* para sincronização e pela transmissão confiável das informações. Existem dois métodos de acesso ao canal: comunicação com *beacon* habilitado e comunicação com *beacon* não habilitado.

No primeiro o coordenador da rede envia periodicamente quadros por *broadcast* para sincronizar os dispositivos. Neste modo não há a necessidade de se utilizar o protocolo CSMA/CA, uma vez que é garantido um intervalo de tempo (GTS) para transmissão para cada dispositivo. No segundo é necessário verificar se o canal está livre antes de se iniciar a transmissão e para isto faz-se o uso do protocolo CSMA/CA.

Estes dois modos refletem diretamente na configuração dos módulos XBee. Isso acontece pois o primeiro permite que o dispositivo esteja em modo *sleep* para economia de energia, ao passo que o segundo não permite tal função e por isso é apenas utilizado nos módulos coordenadores e roteadores.

A subcamada MAC fornece também o algoritmo de AES (Advanced Encryption Standart) para proteger os quadros transmitidos. Além disso, o ZigBee ainda conta com as camadas superiores para garantir a segurança das informações.

2.2.3 Camada de Rede

A camada de rede segundo o modelo OSI é responsável pelo roteamento dos pacotes. No protocolo ZigBee, além desta função, ela possui também outras responsabilidades. Segundo (KINNEY, 2003) a camada de rede deve:

- Detectar e estabelecer com sucesso uma nova rede;
- Associar ou desassociar dispositivos de uma rede;
- Configurar um novo dispositivo pelo *stack profile*;
- Endereçar dos dispositivos que estão entrando na rede pelo dispositivo coordenador.
- Sincronizar dos dispositivos através de *beacons* ou *polling*;
- Assegurar a segurança dos quadros transmitidos/recebidos.

2.2.4 Camada de Aplicação

Segundo o modelo OSI, ela é responsável pelo fornecimento de serviços para as aplicações. A última camada desta pilha é dividida em três componentes, a saber:

- Subcamada de suporte a aplicação (APS): fornece os serviços de descoberta - que determina quais outros dispositivos estão operando na área pessoal de um dispositivo - e o serviço de *binding* - responsável pela vinculação de dispositivos baseados em seus serviços e necessidades e também por encaminhar as mensagens entre eles.
 - *Application Framework*: ambiente onde são alocados até 240 objetos da aplicação ZigBee. Cada objeto representa um *profile* que permite que as aplicações enviem comandos e processem requisições de forma transparente.
 - *ZigBee Device Object*: define o *profile* do dispositivo na rede (ZigBee coordinator, router ou end device), inicializa e/ou responde as requisições de *binding* e estabelece relações seguras entre os dispositivos da rede.

2.2.5 Tipos de dispositivos ZigBee

Para compreender a estrutura de uma rede ZigBee, deve-se explicar os três tipos de dispositivos existentes: *Coordinator*, *Router* e *End Device*.

O ZigBee *Coordinator* é o nó central da rede, sendo único por rede. É o responsável pela formação e gerenciamento da mesma e age também como roteador das mensagens, além de determinar o PAN ID a ser utilizado por todos os dispositivos que fazem parte da mesma rede.

O ZigBee *Router* tem a função de rotear as mensagens pela rede, conectando o coordenador aos dispositivos finais. Este componente, apesar de opcional é de grande importância, uma vez que estende a rede e amplia o alcance da mesma.

O ZigBee *End Device* é o nó final da rede e tem permissão somente para conectar-se a outros *Routers* ou diretamente ao *Coordinator* para troca de mensagens. Ele pode ser configurado para consumir pouca energia por meio do modo *sleep*, sendo o único dos três que possui essa opção. Os outros dois possuem *buffers* para guardar as mensagens que devem ser repassadas aos dispositivos finais caso estes estejam em modo *sleep* no momento do envio. Por este motivo, eles devem estar sempre ativos.

2.2.6 Topologias de Rede

Segundo a norma IEEE 802.15.4, as topologias suportadas são ponto-a-ponto e estrela. Devido a esta restrição, o protocolo ZigBee adicionou novas camadas que permitem redes *multi hop* e *mesh*. Elas serão abordadas com mais detalhes posteriormente.

A topologia em estrela, vista na figura 2.4, possui um coordenador e outros dispositivos que se comunicam diretamente com ele. Assim, todas as mensagens são enviadas para o coordenador, que é o responsável pela posterior distribuição para os outros nodos da rede. Esta topologia possui baixa flexibilidade e por isso é recomendada para projetos de baixa complexidade.

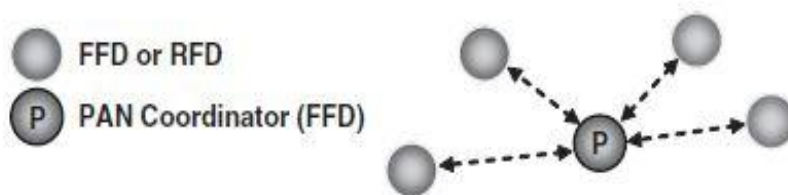


Figura 2.4: Topologia em estrela (FARAHANI, 2008).

A topologia ponto-a-ponto permite dois arranjos: em árvore e em malha. Na rede em árvore há restrições na comunicação entre os dispositivos, isto é, nem todos os roteadores comunicam-se diretamente.

Para compreender melhor este cenário a figura 2.5 mostra uma situação em que o nodo A deseja enviar uma mensagem para o nodo B. Os dois são dispositivos finais, portanto não podem trocar mensagens diretamente. No entanto, não há um roteador que os conecte e por isso a mensagem segue o caminho mostrado na figura:

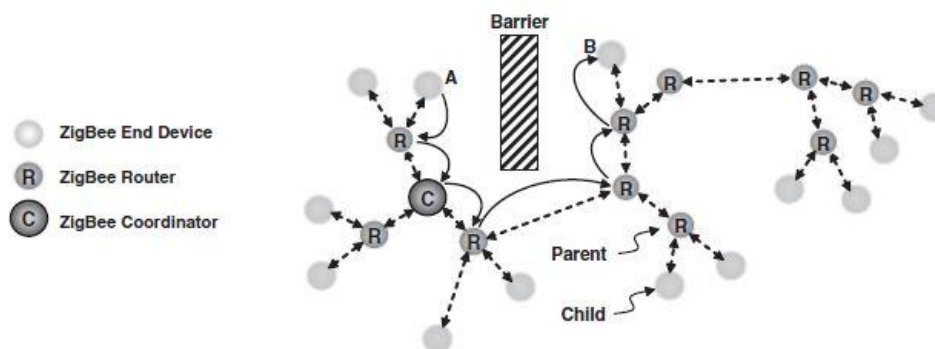


Figura 2.5: Topologia em árvore (FARAHANI, 2008).

Em contrapartida, a rede em malha permite que todos os roteadores comuniquem-se diretamente, fator que permite que novas rotas possam ser definidas caso algum dispositivo deixe ou entre na rede, tornando-a muito flexível. Esta topologia é mostrada na figura 2.6.

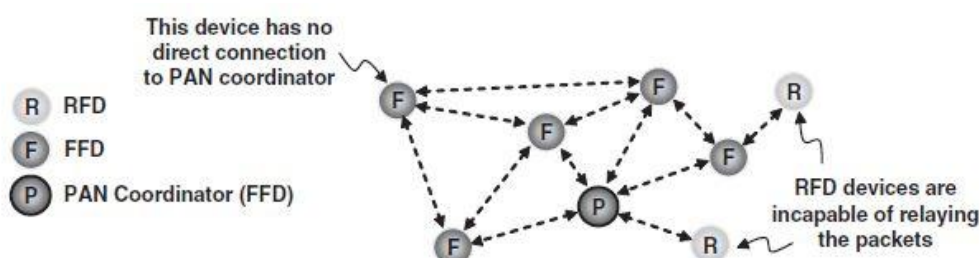


Figura 2.6: Topologia em malha (FARAHANI, 2008).

2.3 O módulo XBee

O XBee é um módulo de rádio, fabricado pela Digi International e suporta uma variedade de protocolos de comunicação, entre eles o ZigBee.

Existem diversas combinações que envolvem diferentes hardwares, firmwares, potência de transmissão e antenas. Neste trabalho foi utilizado o XBee Series 2 que permite utilizar a rede ZigBee em malha. Este modelo possui onze pinos digitais de entrada e saída, sendo que quatro destes pinos podem ser utilizados no modo analógico. A figura 2.7 mostra o módulo XBee utilizado neste trabalho.



Figura 2.7: XBee Series 2¹

¹ Imagem retirada de Adafruit Industries: <http://www.adafruit.com/product/968>. Acesso em 06/06/2014.

Para a configuração dos módulos XBee existem dois modos de operação: por API e por comandos AT. Ao utilizar o programa X-CTU, desenvolvido pela Digi International exclusivamente para configuração dos XBees, é possível escolher em qual destes dois modos o módulo irá operar. Mais detalhes sobre o programa e as configurações dos módulos serão abordados no capítulo 3.

O modo por comandos AT é mais simples e é mais indicado para conexões ponto-a-ponto. Para este trabalho foi utilizado o modo API (Application Programming Interface) que faz uso do envio e recebimento de *frames*. Assim é possível analisar a estrutura dos *frames* e obter as informações necessárias para o sistema de automação residencial implementado, além de ser uma forma altamente confiável de transmissão de dados. Além disso, também é possível enviar comandos AT via modo API para configuração remota dos módulos XBee.

Start delimiter	Length		Frame data			Checksum
Byte 1	Byte 2	Byte 3	Byte 4	...	Byte n	Byte n+1
0x7E	MSB	LSB	API-specific structure			Single byte

Figura 2.8: Estrutura do quadro no modo API (FALUDI, 2011)

Analisando a figura 2.8 é possível identificar os seguintes campos:

- *Start delimiter*: byte que identifica o início do *frame*;
- *Length*: 2 bytes que informam o tamanho do campo de dados;
- *Frame data*: este campo é composto pelo tipo de *frame* (*Remote Request*, *Remote responde*, entre outros) e por diversos outros campos que variam conforme o tipo escolhido;
 - *Checksum*: byte para verificar a integridade do *frame*, calculado a partir do *Frame data*.

Figura 2.9: Exemplo de um quadro API

A figura 2.9 apresenta um exemplo de um tipo de *frame*. Ele foi gerado com o auxílio da ferramenta *Frame Generator* presente no software X-CTU. O tipo escolhido foi o *Remote AT command*.

Como pode-se perceber, além dos campos explicados anteriormente, observa-se a presença do *Frame ID*, utilizado na identificação e verificação de perdas ou erros na transmissão. Neste caso foi usado o ID zero que suprime qualquer tipo de resposta pelos módulos XBee. Logo após o *Frame ID*, observam-se dois campos de endereço: o primeiro é o endereço único de cada módulo que quando utilizado, exclui a necessidade de utilização do outro campo de endereçamento. O segundo campo é utilizado para endereçamento de redes e não foi usado neste trabalho. Em seguida há um campo de opções (1 byte), acompanhado do campo com o comando que deseja-se enviar (2 bytes) e por fim o valor atribuído a este comando. Após isto, calcula-se o *checksum* e o adiciona ao *frame*. Dentro do retângulo vermelho pode-se visualizar o *frame* completo.

A lista completa dos comandos encontra-se no site da Digi International ou por meio do software X-CTU.

3 IMPLEMENTAÇÃO DO PROJETO

O sistema de automação residencial implementado neste trabalho aborda algumas das possibilidades existentes para controle de dispositivos por meio de uma rede sem fio. No decorrer deste capítulo serão mostrados e explicados todos os componentes utilizados e como a integração entre eles ocorre.

O projeto foi desenvolvido em quatro partes, definidas da seguinte maneira: a primeira parte é constituída pelo hardware, onde serão explicados os circuitos. A segunda é constituída pelo aplicativo Android, a terceira pelo software do módulo controlador e a última parte consiste na incorporação da conectividade com a Nuvem.

3.1 Arquitetura do Sistema

O sistema foi desenvolvido em função dos protocolos de comunicação escolhidos e para compreender o seu funcionamento uma explicação inicial sobre a arquitetura é necessária.

A figura 3.1 mostra a arquitetura utilizada neste trabalho e os principais componentes que fazem parte do sistema. Em um momento inicial o usuário acessa o aplicativo por meio de um dispositivo móvel para controle dos sensores. Ao selecionar uma ação, o comando correspondente é enviado por Wi-Fi para o módulo controlador, que o repassa para os módulos que possuem os sensores via protocolo ZigBee.

Os detalhes de implementação de cada componente serão descritos nos capítulos a seguir, tanto em nível de hardware quanto em nível de software.



Figura 3.1: Arquitetura do sistema de automação residencial

3.1.1 Arquitetura de Hardware

Antes de detalhar cada circuito, será mostrado como ocorre a conexão física entre os diferentes componentes que fazem parte deste sistema e para melhor compreensão do que será explicado, a figura 3.2 mostra as conexões de todo o sistema.

Como já mencionado, o dispositivo móvel comunica-se com o módulo controlador por meio do protocolo Wi-Fi. Este módulo é constituído primordialmente pela placa Arduino e por um XBee Coordenador, que comunicam-se via serial. Os demais componentes deste módulo estão conectados diretamente nos pinos I/O do próprio Arduino e serão mostrados no capítulo 3.2.1.

Em seguida, a comunicação entre os rádios XBee ocorre por meio do protocolo ZigBee. Nos circuitos que possuem os sensores, eles estão conectados aos pinos I/O dos XBees, que possuem configurações diferentes para atuar de formas diferentes. Estes detalhes serão abordados no capítulo seguinte, que trata da arquitetura do software que foi implementado.

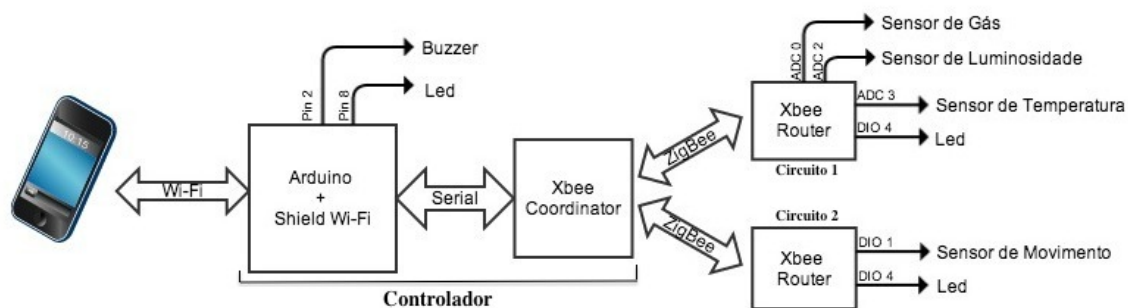


Figura 3.2: Diagrama que mostra as conexões em nível de hardware

3.1.2 Arquitetura de Software

A implementação de software é composta de dois módulos: um deles usado no dispositivo com Android e o outro no controlador.

O módulo do Android foi desenvolvido em Java. Neste projeto existem duas atividades, que caracterizam cada uma das telas que o usuário pode interagir.

A primeira tela possui somente um botão que dá acesso a próxima atividade e que também é responsável por verificar se há conexão Wi-Fi. Caso haja, a segunda tela aparece e nela há diversos botões.

Seguindo o diagrama da figura 3.3 para se ter acesso aos demais botões, primeiro deve-se apertar o botão *Connect*, este botão é responsável por criar a conexão com o servidor, permitindo a comunicação entre o dispositivo e o controlador.

Os botões referentes aos sensores, possuem duas opções: ligar ou desligar. Primeiro verifica-se qual o valor do botão e em seguida, inverte-se esse valor. Então a informação referente ao botão que foi apertado é enviado pelo socket. Além disso, há o botão para atualizar o valor da temperatura, que não altera nenhum valor e só envia a informação pelo socket.

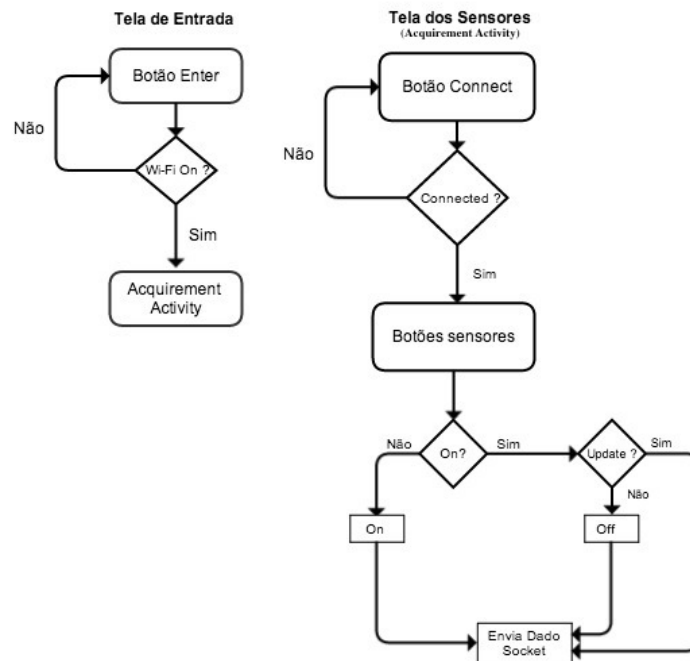


Figura 3.3: Arquitetura do aplicativo Android

Para auxiliar a compreensão da implementação do controlador, o diagrama da figura 3.4 mostra o fluxo de funcionamento do código.

Após a inicialização, verifica-se se há clientes disponíveis. Caso não haja, a função para análise de frames é chamada. Esta função é responsável por verificar se existem *frames* que foram enviados pelos sensores e analisá-los se for o caso.

Nesta função de análise, os campos dos *frames* recebidos são verificados. Dependendo do seu valor, identifica-se que determinado sensor enviou a informação e ela deve ser tratada de acordo. Caso seja o sensor de movimento, identifica-se movimento e o alarme será disparado. Se for o sensor de gás, identifica-se vazamento. O sensor de luminosidade informa a intensidade da luz no ambiente e o sensor de temperatura informa a temperatura ambiente.

Se não houver *frames* para serem analisados a função termina. Seguindo o fluxo do código, novamente verifica-se se há clientes disponíveis. Supondo que haja, o próximo passo é verificar se ele está conectado. Se não estiver, o socket é finalizado e o processo se repete. Se estiver, mais uma vez a função que analisa os *frames* é chamada. Terminada a execução da função, verifica-se se o cliente está enviando informações pelo socket.

Ao constatar que o cliente está enviando informações, a primeira comparação é para verificar se é um comando para ligar/desligar algum sensor. Se for um destes comandos, identifica-se qual o sensor que será acionado e são chamadas as funções referentes ao led, ao envio do *frame* para o sensor destino, ao buzzer e ao envio da informação para a plataforma na Nuvem.

Esta última função estabelece uma conexão com o servidor da plataforma e então envia a informação no formato JSON por meio de uma requisição HTTP. Esta requisição também é montada, semelhante ao *frame* enviado para os sensores, e pode

ser visualizada em mais detalhes diretamente no código, na função `SendData`, a partir da linha 1209.

As demais funções são explicadas no capítulo 3.3.2, que aborda em mais detalhes o módulo controlador.

Se o comando solicitado é para atualizar a informação da temperatura, ele somente chama a função que monta e envia o *frame* para o sensor em questão.

Por fim, quando o cliente não estiver mais conectado, o socket é fechado e todo o ciclo se repete.

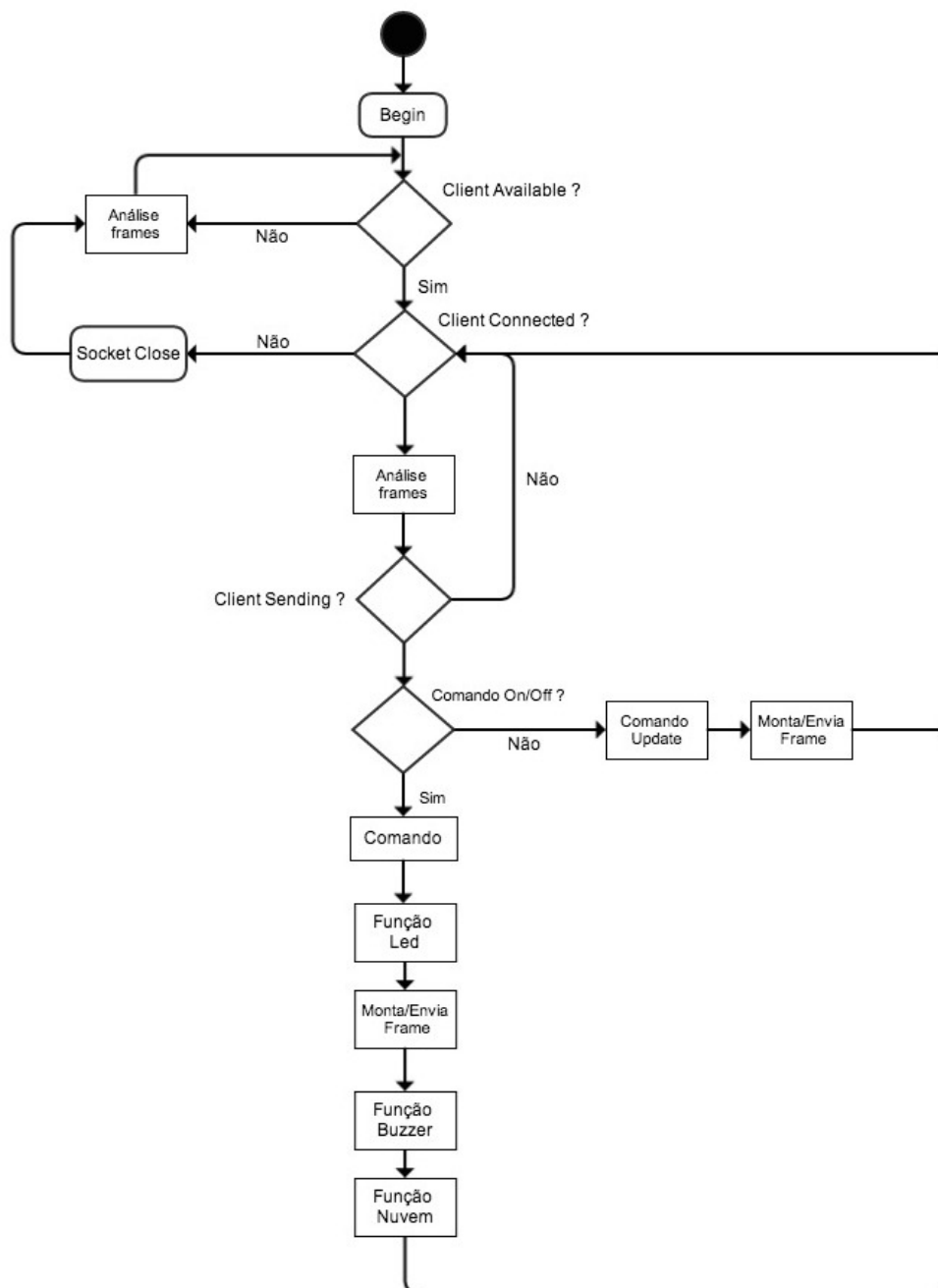


Figura 3.4: Arquitetura do controlador

3.2 Circuitos e sensores

Para este trabalho foram contruídos três circuitos. O primeiro acoplou-se junto à placa Arduino e compõe o módulo controlador, ao passo que os outros dois são utilizados em conjunto com os sensores. A proposta inicial previa um circuito por sensor, mas devido à limitação de módulos XBee trabalhou-se apenas com dois circuitos que serão detalhados posteriormente.

3.2.1 Controlador

Este módulo é composto pela placa Arduino Mega 2560, pelo shield Wi-Fi CC3000 produzido pela Adafruit Industries, por um módulo XBee (em conjunto com um adaptador) configurado como coordenador e um circuito auxiliar. O circuito foi construído utilizando-se uma protoboard e pode ser visualizado na figura 3.2.

Na protoboard acoplou-se o XBee com o adaptador que, por sua vez, possui uma entrada USB mini que permite conectar o módulo diretamente ao computador para configuração por meio do software X-CTU. Uma outra função deste adaptador é converter o espaçamento dos pinos do XBee para um compatível aos da protoboard. Além disso foram utilizados alguns resistores, um led e um buzzer que indicam o (des)acionamento dos sensores.

O Arduino é alimentado por uma fonte de 12V e possui um pino de saída com tensão regulada em 3,3V, utilizado para alimentar o XBee. As demais conexões foram:

- Pino *GND*, cuja função foi o aterramento da protoboard e dos demais components;
- Pino 2, ligado ao buzzer;
- Pino 8, ligado ao led;
- Pinos 18 e 19, Tx e Rx respectivamente, conectados ao Rx e Tx do XBee para comunicação via serial número 1.

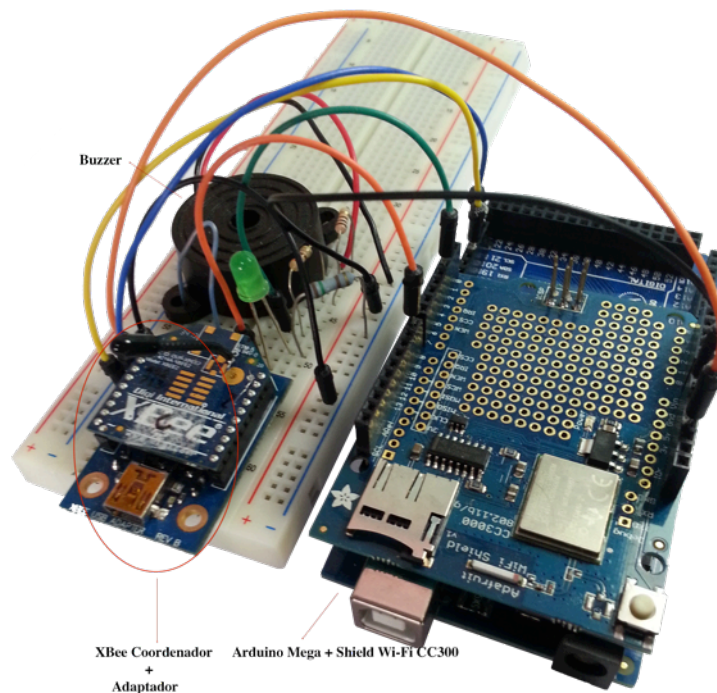


Figura 3.5: Circuito Controlador

3.2.2 Sensores

3.2.2.1 Movimento

Foi utilizado no circuito um sensor que detecta variações de radiação infravermelha. Estas variações ocorrem quando uma pessoa ou animal passa em torno do perímetro coberto pelo sensor, identificando a ocorrência do movimento. Eles são de baixo custo e consomem pouca energia, o que os torna ideal para este trabalho.

O seu funcionamento consiste no seguinte mecanismo: quando uma alteração é detectada, os sinais analógicos captados são convertidos para digitais, fornecendo apenas dois valores: zero ou um. Neste projeto o sinal é enviado para uma das portas digitais do XBee, configurada para monitorar mudanças nos valores dos pinos.

Na protoboard foi acoplado o XBee com o adaptador, alguns resistores, dois leds, o sensor de movimento e uma placa para conectar uma fonte para alimentação da protoboard e dos componentes. O adaptador é alimentado com 5V, porém há a necessidade de um regulador de tensão de 3,3V para que o módulo XBee seja devidamente alimentado. O circuito é mostrado na figura 3.3.

As conexões utilizadas estão explicitadas abaixo

- Pino DIO1: entrada digital, configurada para detectar transições *HIGH to LOW* ou *LOW to HIGH*, indicando uma variação IR;
- Pino DIO4: entrada digital ligada a um led vermelho. Este led serve como auxiliar visual para identificar o (des)acionamento do sensor;
- Pinos VDD e GND: pinos do adaptador para alimentar o XBee;

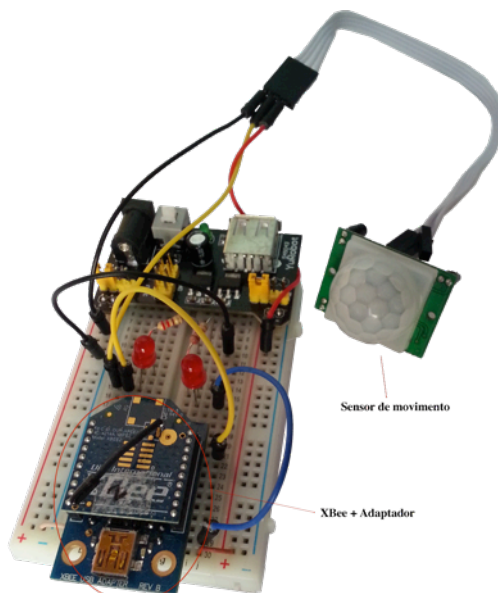


Figura 3.6: Circuito com sensor de movimento

3.2.2.2 Gás

Para este circuito foi utilizado o sensor MQ-2, capaz de detectar a presença de vários gases, entre eles o butano, conhecido popularmente como gás de cozinha.

Por se tratar de um sensor analógico, os sinais são enviados para uma das portas ADC do módulo XBee. No entanto, como essas portas funcionam com uma tensão máxima de 1,23V, é necessário um divisor de tensão, como mostrado na figura 3.4.

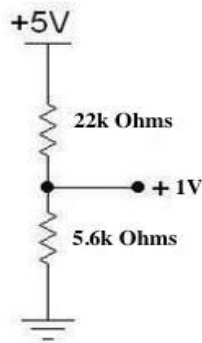


Figura 3.7: Divisor de tensão

Devido a uma limitação no número de componentes, não foi possível montar um circuito por sensor, sendo utilizada apenas uma protoboard para este e os próximos sensores.

Com o auxílio do Fritzing, software para desenho de circuitos elétricos, foi possível visualizar como seria o circuito se construído individualmente.

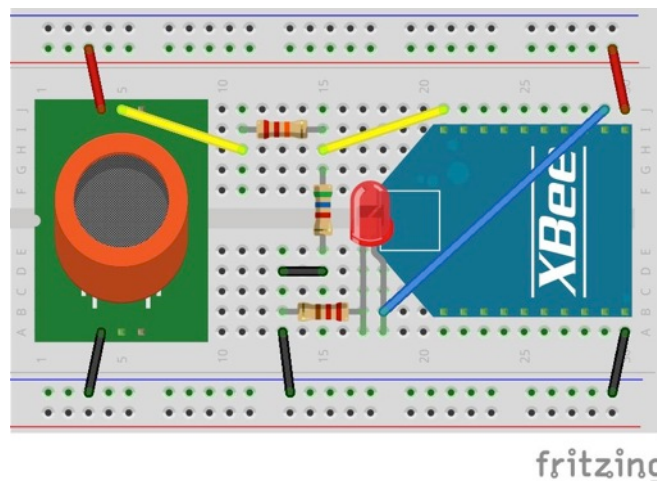


Figura 3.8: Desenho do circuito com sensor de gás

As conexões do circuito apresentado na figura 3.5 são semelhantes ao anterior, alterando-se somente o pino digital pelo analógico.

3.2.2.3 Luminosidade

Este sensor é um fotoresistor que altera a sua resistência dependendo da quantidade de luz que incide sobre ele. Eles são pequenos e econômicos e são em geral indicados para determinar mudanças na incidência de luz, sem no entanto informar valores exatos.

Neste circuito também trabalhou-se com sinais analógicos e, portanto, houve a necessidade de um divisor de tensão. Como o próprio sensor possui resistência variável, a escolha do outro resistor deu-se a partir de medições e cálculos, de modo a manter a tensão sempre abaixo de 1,23V.

Novamente o circuito da figura 3.6 é semelhante aos anteriores, alterando-se somente a porta analógica, que passou a ser ADC2 e pôde também ser idealizado com o auxílio do software Fritzing.

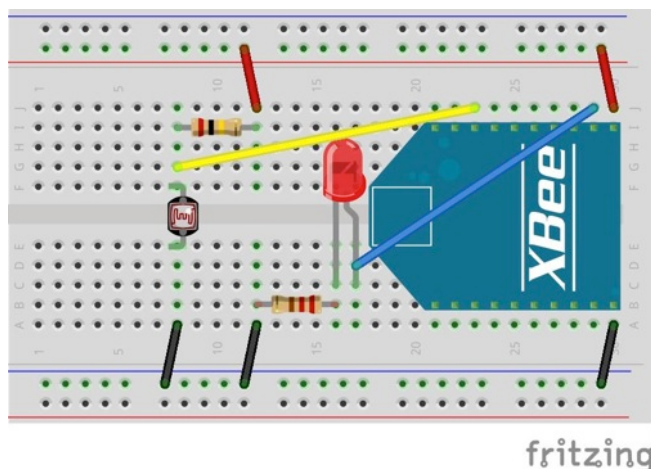


Figura 3.9: Desenho do circuito com sensor de luz

3.2.2.4 Temperatura

O sensor de temperatura escolhido foi o LM35, fabricado pela National Semiconductor, que fornece uma tensão linear proporcional à temperatura que se encontra (em graus Celsius).

Neste circuito há também um divisor de tensão, porém ele só é necessário quando as temperaturas medidas estão acima dos 100°C . Como as medições são limitadas às temperaturas internas de uma residência - que não costumam atingir estas temperaturas -, o pino V_{out} do sensor está conectado diretamente na porta analógica ADC3 do Xbee.

O desenho do circuito, novamente feito com o auxílio do software Fritzing, é mostrado na figura 3.7. Na figura 3.8 é apresentado o circuito montado com os sensores de gás, luminosidade e temperatura.

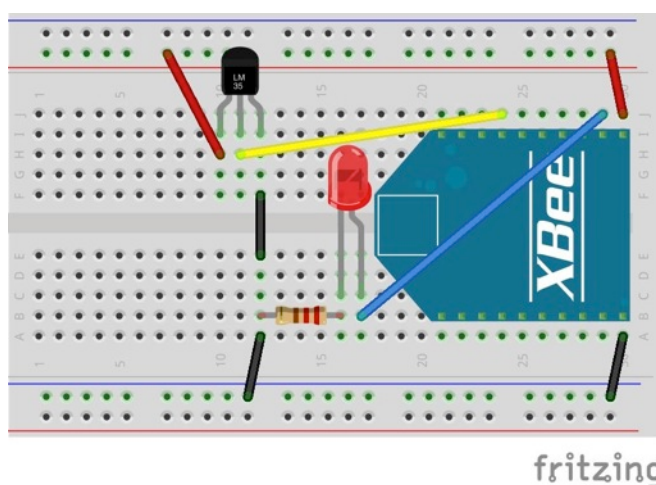


Figura 3.10: Desenho do circuito com sensor de temperatura

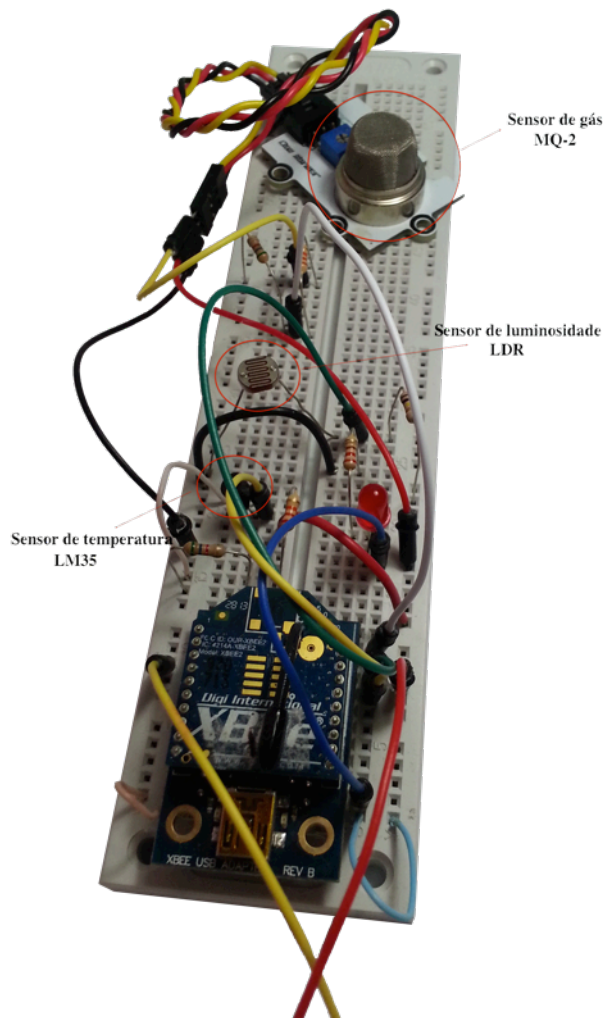


Figura 3.11: Circuito único com os sensores de gás, luminosidade e temperatura

3.3 O Sistema de Automação

A programação do sistema de automação foi dividida em três fases: programação do aplicativo móvel para Android, programação do controlador e integração dos dados com a Nuvem. Além disso, foi necessário configurar os módulos XBee utilizando o software X-CTU.

3.3.1 Aplicativo Android

A ampla utilização de dispositivos móveis como celulares e tablets impulsiona a criação de novos aplicativos para auxiliar as pessoas nas mais diversas atividades. Por este motivo, foi desenvolvido o Smart House Control, aplicativo para automação residencial para a plataforma Android. A escolha deste sistema operacional se deu pela disponibilidade de celulares para instalação e possibilidade de testes do aplicativo.

O desenvolvimento foi dividido em duas etapas: programação da interface gráfica e programação para comunicação via sockets.

A interface gráfica foi programada utilizando a linguagem XML, padrão para desenvolvimento dos aplicativos Android. Esta parte não envolveu nenhuma solução

especial, apenas posicionamento de elementos gráficos nas telas. O aplicativo com a interface final pode ser visualizado na figura 3.9.

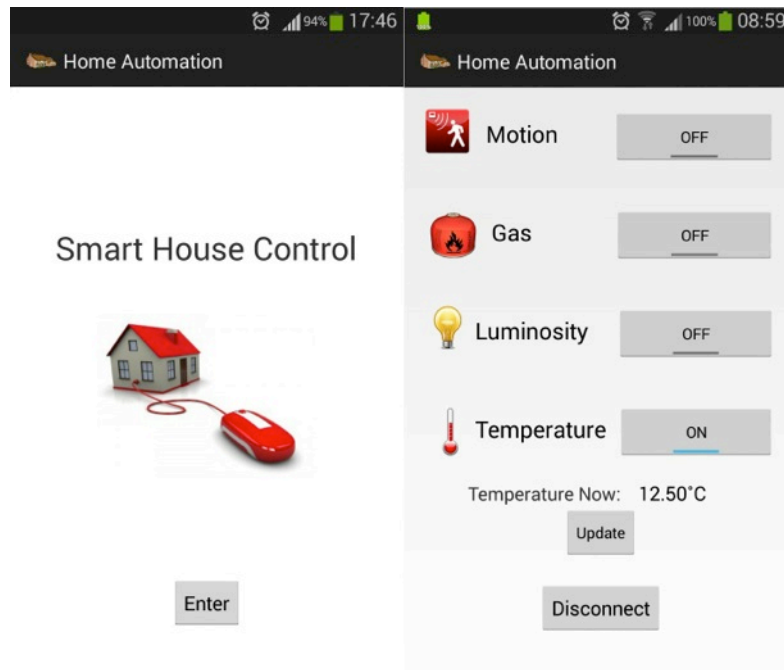


Figura 3.12: Tela de entrada e tela secundária

A segunda parte envolveu a programação da comunicação entre o Android e o controlador. Para isto decidiu-se pelo formato cliente/servidor, sendo o Android o cliente e o controlador, o servidor.

Do lado do cliente optou-se por utilizar as *Tasks* por meio da classe *AsyncTask*, que permite o uso apropriado da *User Interface Thread*. Assim é possível fazer operações em *background* e enviar os resultados para a tela sem a necessidade de lidar diretamente com *threads* ou *handlers*.

Essa classe possui cinco métodos implementados:

- *onPreExecute*: chamada antes da *task* iniciar;
- *doInBackground*: este método inicia uma conexão via sockets e a deixa aberta enquanto não for finalizada;
- *onProgressUpdate*: o método é chamado para atualizar a tela ao receber uma nova string;
- *onCancelled*: chamada quando a *task* é cancelada;
- *onPostExecute*: chamada após a execução da *task*;

3.3.1.1 Smart House Control

Após iniciar o aplicativo e apertar o botão *Enter*, a tela secundária da figura 3.9 aparece. Nesta tela o usuário deve primeiramente apertar o botão *Connect*, para estabelecer a conexão com o controlador (servidor). Caso isso não seja feito, uma mensagem de alerta aparecerá para o usuário. Após a conexão, o usuário pode escolher quais sensores deseja (des)acionar e solicitar atualização da temperatura ambiente.

Ao apertar o botão desejado, uma string é enviada pelo socket, sendo depois tratada apropriadamente no controlador. Ao solicitar a atualização do valor da temperatura

observa-se o retorno de uma string pelo socket. Depois, o método `onProgressUpdate` é chamado e o valor é atualizado na tela. Ao terminar de utilizar o aplicativo, o usuário desconecta do servidor e sai do aplicativo.

3.3.2 Módulo Controlador

O controlador é o componente principal deste sistema de automação residencial e atua como um servidor, responsável por receber e transmitir informações tanto para o cliente Android quanto para os sensores. Ele foi programado em linguagem C utilizando a IDE do Arduino.

Ao receber os comandos via socket, cada um deles é tratado para realizar uma ação diferente. Essas ações seguem o seguinte padrão:

- Reconhecer o comando;
- Chamar função para (des)ligar o led presente no circuito;
- Chamar a função para (des)acionar o sensor;
- Chamar a função que aciona o Buzzer;
- Chamar a função para enviar dados para a Nuvem.

Cada circuito possui um led cuja função é indicar se o sensor foi acionado ou não. O que esta função faz é montar e enviar um *frame* com o valor desejado para a porta na qual o led está conectado. Enviando o valor hexadecimal 0x04 desliga-se o led e enviando o valor hexadecimal 0x05 liga-se o led.

A função para (des)acionar os sensores funciona de forma semelhante, porém para cada sensor é montado e enviado um *frame* diferente, com os valores e configurações desejados. A título de exemplo de como isto foi realizado no código, no trecho a seguir são enviados dois *frames*: o primeiro ativa o pino com o sensor de movimento e o segundo configura o XBee para detectar alterações do tipo *high/low* no pino do sensor.

```

//Packet 2
// Begin the API frame
sendByte(frameStartByte);
// High and low parts of the frame length (not counting checksum)
sendByte(0x0);
sendByte(0x10);
// Accumulate the checksum
long sum = 0;
// Indicate this frame contains a Remote AT command
sum += sendByte(frameTypeRemoteAT);
// frame ID set to zero for no reply
sum += sendByte(0x0);
// The following bytes indicate the 64-bit address of the recipient.
// DH: 0x0013A200
sum += sendByte(0x0);
sum += sendByte(0x13);
sum += sendByte(0xA2);
sum += sendByte(0x0);
// DL: Depends on the sensor
sum += sendByte(sensorAddress0);
sum += sendByte(sensorAddress1);
sum += sendByte(sensorAddress2);
sum += sendByte(sensorAddress3);

// The following 2 bytes indicate the 16-bit address of the recipient.
// Not used in this case
sum += sendByte(0xFF);
sum += sendByte(0xFE);

// Send Remote AT options
sum += sendByte(remoteATOptionApplyChanges);

// The text of the AT command
sum += sendByte('D');
sum += sendByte('1');

// The value (0x03 for digital input, 0x0 for disable)
sum += sendByte(0x03);

// Send the checksum
sendByte(0xFF - (sum & 0xFF));

//Packet 3 - IC
sendByte(frameStartByte);
sendByte(0x0);
sendByte(0x10);

sum = 0;

sum += sendByte(frameTypeRemoteAT);
sum += sendByte(0x0);
// DH: 0x0013A200
sum += sendByte(0x0);
sum += sendByte(0x13);
sum += sendByte(0xA2);
sum += sendByte(0x0);
// DL: Depends on the sensor
sum += sendByte(sensorAddress0);
sum += sendByte(sensorAddress1);
sum += sendByte(sensorAddress2);
sum += sendByte(sensorAddress3);

sum += sendByte(0xFF);
sum += sendByte(0xFE);

// Send Remote AT options
sum += sendByte(remoteATOptionApplyChanges);

// The text of the AT command
sum += sendByte('I');
sum += sendByte('C');

// The value (0x0F mask for D3, D2, D1, D0)
sum += sendByte(0x0F);

// Send the checksum
sendByte(0xFF - (sum & 0xFF));

```

Figura 3.13: Trecho do código de acionamento do sensor de movimento

Analisando-se os dois trechos mostrados na figura 3.10 é possível perceber que o primeiro byte indica o início do *frame* (*frameStartByte* - com o valor hexadecimal 0x7E). Os próximos dois bytes fornecem o tamanho do *frame data*. Logo após é declarada uma variável *sum* para cálculo do *checksum*. Em seguida é definido o tipo do *frame* (*frameTypeRemoteAT*), que será um comando AT remoto para configuração do XBee. O byte seguinte informa que o ID é zero e portanto o XBee remoto não necessita enviar *frames* de resposta (ou ACKs). Os 8 bytes seguintes são para endereçamento do XBee remoto - cada sensor possui um endereço único - e mais dois bytes para endereçamento de redes, caso não abordado neste trabalho. A seguir, observa-se o byte de opções (*remoteATOptionApplyChanges*) que informa que todas as configurações no módulo XBee devem ser aplicadas imediatamente. Finalmente, observam-se os campos com o comando AT e o valor que deve ser aplicado em cada pino. No primeiro trecho informa-se que o pino D1 será uma entrada digital e no segundo informa-se que as portas D0, D1, D2 e D3 serão monitoradas para detectar alterações. Por fim, tem-se o byte de *checksum*.

A função que aciona o buzzer é responsável por emitir dois sinais sonoros para diferenciar quando o sensor é ligado ou desligado, seguindo o mesmo raciocínio aplicado aos leds. Por último há a função de integração com a Nuvem que será abordada em detalhes no próximo item.

Uma parte muito importante do código do controlador é aquela responsável por analisar os *frames* que foram enviados pelos sensores. Com isso é possível identificar quais as informações são pertinentes para o funcionamento do sistema de automação residencial e aplicar as ações correspondentes. Por exemplo, ao analisar o *frame* enviado pelo sensor de detecção de gás, verifica-se o valor lido do pino. Caso o valor seja maior que uma constante (determinada por meio de testes com o sensor), o alarme será disparado, avisando para o usuário que houve detecção de vazamento de gás. Situação semelhante ocorre com o sensor de movimento: ao analisar o valor do pino, verifica-se que houve alteração e o alarme é disparado.

Já o sensor de luminosidade verifica em qual intervalo encontra-se o valor lido, identificando a intensidade da luz (muita luz, pouca luz ou escuro), enquanto o sensor de temperatura informa o valor lido em graus Celsius.

É importante salientar que para a obtenção dos valores a partir dos sensores de gás e luz foi aplicado uma máscara, pois os valores analógicos utilizam 10 bits de representação (valores entre 0 e 1023) e por isso são divididos em dois bytes na hora de enviar o dado:

```
int analogTempValue = analogLSB + (analogMSB * 256)
```

O valor recuperado é armazenado em uma variável inteira, que é a soma do byte menos significativo com o byte mais significativo após a aplicação da máscara.

Para o sensor de temperatura esta máscara também é aplicada. No entanto, para obter-se o valor da temperatura ainda é necessária a aplicação da seguinte fórmula:

```
float celsius = ((analogTempValue/1023.0)*1230)/10.0
```

Primeiramente o valor anterior é dividido por 1023, devido a representação de 10 bits utilizada. O restante da fórmula foi retirado do datasheet, que informa que cada grau corresponde a 10mV. Como cada porta analógica só permite uma tensão máxima

de 1,23V, este é o valor utilizado na fórmula. O resultado é armazenado em uma variável do tipo float.

3.3.3 Conectividade com a Nuvem

Além do aplicativo que permite o controle local do sistema de automação residencial, é interessante fornecer mais ferramentas para o usuário acompanhar remotamente o que ocorre em sua residência.

Para integrar os dados com a Nuvem, foi escolhida a plataforma Carriots, que oferece recursos vantajosos e pode ser acessado de qualquer dispositivo que possua conexão com a internet.

Primeiramente é necessário criar uma conta no site:

- <https://www.carriots.com/>

Depois de criada a conta, cadastram-se todos os sensores cujos dados serão enviados para a Nuvem. Na figura 3.11. pode-se visualizar a lista com os sensores cadastrados no site.

Device List

Name	Description	Time zone	Enabled	Status	Actions
Temp		America/Sao_Paulo	✓	●	⌵ Actions
Light		America/Sao_Paulo	✓	●	⌵ Actions
Gas		America/Sao_Paulo	✓	●	⌵ Actions
Motion		America/Sao_Paulo	✓	●	⌵ Actions

OK ● Disconnected ● No_status or no_data ●

< Previous | 1 | Next > Results | 4 |

Figura 3.14: Sensores cadastrados na plataforma Carriots

A terceira parte envolve a programação do controlador para enviar os dados via REST API. Ela foi realizada da seguinte forma:

- A informação do sensor é formatada para ser enviada como um objeto JSON;
- A conexão com o servidor Carriots é estabelecida;
- A requisição HTTP é enviada junto com o API Key (presente no menu "My account");

A partir disso, os dados enviados aparecerão na lista de *data streams*. Um exemplo de como eles são visualizados no site é mostrado na figura 3.12.

Data Stream List

SEARCH			
at	device	data	Actions
08/06/2014 10:02:03	Motion@brunaseewald.brunaseewald	{"Motion": "ALARM!"}	Actions
08/06/2014 10:00:02	Motion@brunaseewald.brunaseewald	{"MotionSensor": "ON"}	Actions
08/06/2014 09:59:18	Gas@brunaseewald.brunaseewald	{"GasSensor": "OFF"}	Actions
07/06/2014 13:55:52	Gas@brunaseewald.brunaseewald	{"GasSensor": "ON"}	Actions
07/06/2014 13:55:44	Temp@brunaseewald.brunaseewald	{"TempSensor": "OFF"}	Actions
07/06/2014 13:55:28	Temp@brunaseewald.brunaseewald	{"Degrees Celcius": 13.94}	Actions
07/06/2014 13:55:14	Temp@brunaseewald.brunaseewald	{"Degrees Celcius": 13.82}	Actions
07/06/2014 13:55:06	Temp@brunaseewald.brunaseewald	{"TempSensor": "ON"}	Actions
04/06/2014 23:34:20	Temp@brunaseewald.brunaseewald	{"TempSensor": "OFF"}	Actions
04/06/2014 17:25:22	Temp@brunaseewald.brunaseewald	{"Degrees Celcius": 11.42}	Actions

< Previous | 1 ... 11 12 13 (14) 15 16 17 ... 53 | Next > Results | 526 |

Figura 3.15: Lista com os dados enviados para a Nuvem

Um recurso muito interessante é a possibilidade de enviar mensagens SMS e/ou emails caso um evento ocorra. Neste trabalho, as duas formas de aviso foram utilizadas, oferecendo maior flexibilidade e praticidade para o usuário.

Na figura 3.13 são mostrados os *Listeners* para envio de email e mensagem SMS atrelados aos sensores de movimento e gás. No exemplo da figura 3.14 a condição para envio é que o alarme tenha disparado e na figura 3.15 há um exemplo de um email que foi recebido pelo usuário.

Listener List

SEARCH					
Name	Description	Event	Entity	Enabled	Actions
ListenerSendSMS2	Listener that sends a SMS	Event Data Received	Gas@brunaseewald.brunaseewald (Device)	✓	Actions
ListenerSendEmail2	Listener that sends an email	Event Data Received	Gas@brunaseewald.brunaseewald (Device)	✓	Actions
ListenerSendSMS1	Listener that sends a SMS	Event Data Received	Motion@brunaseewald.brunaseewald (Device)	✓	Actions
ListenerSendEmail1	Listener that sends an email	Event Data Received	Motion@brunaseewald.brunaseewald (Device)	✓	Actions

< Previous | 1 | Next > Results | 4 |

Figura 3.16: Listeners que enviam mensagens para alertar o usuário

Name	ListenerSendEmail1	Id developer	ListenerSendEmail1@brunaseewald.brunaseewald
Entity type	Device	Event	Event Data Received
Id	Motion@brunaseewald.brunaseewald		
If expression	1 context.data.Motion == "ALARM!"		
Then expression	<pre> 1 import com.carriots.sdk.utils.Email; 2 3 def email = new Email(); 4 email.to="brubetinha@gmail.com"; 5 email.subject="Motion Alarm"; 6 email.message="Alarm activated!"; 7 8 email.send(); </pre>		

Figura 3.17: Listener para enviar email

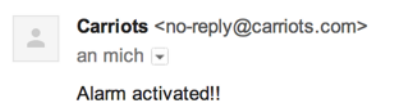


Figura 3.18: Email alertando o usuário sobre o alarme ativado

O código fonte do projeto está disponível no site GitHub. O acesso aos arquivos é liberado para todos e os projetos são disponibilizados por meio da licença Creative Commons BY.

O endereço para acesso é:

- <https://github.com/brubetinha/home-automation-project>

3.3.4 Configuração dos módulos XBee

Cada módulo possui uma série de parâmetros que devem ser configurados por meio do software X-CTU antes da utilização no sistema de automação residencial.

Ao conectar os XBees via USB deve-se selecionar a entrada correta e as configurações a seguir: *baud rate* de 115200, 8 data bits, sem bit de paridade, 1 stop bit e sem *flow control*. Essas opções não são fixas e podem ser alteradas de acordo com a necessidade do projeto. Todos os módulos pertencem à família XB24-ZB.

Um dos principais parâmetros que deve ser configurado via software é o PAN ID, definido pelo coordenador e que deve ser o mesmo para todos os dispositivos da rede. Este parâmetro e outros são mostrados na figura 3.16.

A função de cada módulo e a versão do firmware também são gravadas no programa. O coordenador possui a função *ZigBee Coordinator API*, que permite o modo de operação API e a transmissão dos dados por *frames*. Os demais módulos podem ser configurados com as funções *ZigBee End Device AT* ou *ZigBee Router AT*. Como já mencionado, *end devices* permitem que os dispositivos sejam configurados em modo *sleep* e os *routers* permitem a extensão da rede. Neste trabalho utilizou-se a segunda opção para os outros módulos XBees.

As demais configurações foram realizadas dinamicamente por meio do envio de *frames* do tipo *Remote AT Command*.

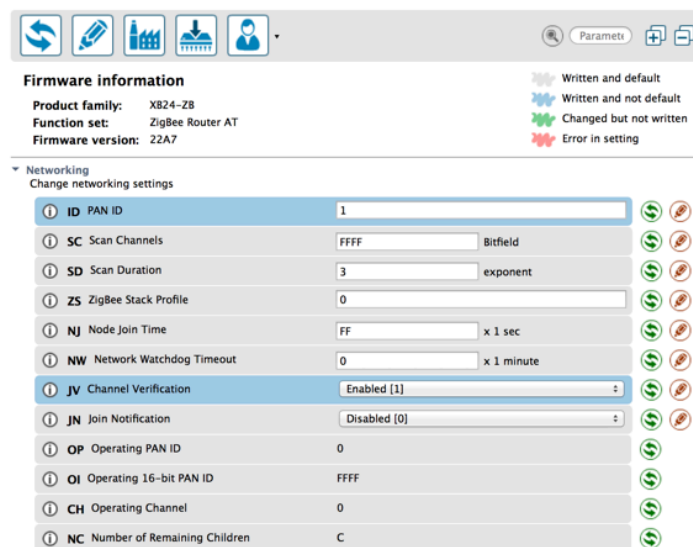


Figura 3.19: Tela do software X-CTU

3.4 Custo total do projeto

Este projeto baseou-se fortemente no custo dos componentes para ser realizado e por isso é importante ter conhecimento do preço dos itens mais importantes.

Alguns componentes foram importados dos Estados Unidos, portanto o seus preços serão informados em dólares. Outros itens custaram poucos centavos e não serão incluídos na listagem abaixo:

- 3x XBee Series 2: preço unitário \$ 22,95, total \$ 68,85
- 3x USB XBee Adapter: preço unitário \$ 20,00, total \$ 60,00
- Adafruit CC3000 WiFi Shield : \$ 39,95
- Sensor de movimento PIR DYP-ME003: R\$ 12,90
- Sensor de gás MQ-2: R\$ 19,90
- Sensor de temperatura LM35: R\$ 6,50
- Placa Arduino Mega 2560: R\$ 69,90
- Fontes de tensão de 12V e 9V : R\$ 41,00

Com o dólar cotado a R\$ 2,30, o preço total dos itens importados foi R\$ 388,47. Somando este valor ao restante:

- Custo do projeto: R\$ 538,67

Este valor não inclui gastos com frete nem imposto de importação. Considerando estas despesas foram gastos mais R\$ 308,00, totalizando o valor em:

- Custo total: R\$ 846,67

Ao explicitar os preços dos componentes observa-se que o fator preponderante para os valores cobrados em serviços de automação é oriundo da mão de obra altamente especializada requerida. No entanto, mesmo com a utilização de uma mão de obra com alto valor agregado, os preços encontrados no mercado – uma automação residencial completa custa a partir de R\$ 15.000,00 – são muito elevados.

Esta discrepância entre o valor de mercado e aquele proposto pelo trabalho pode ser oriunda de diversos fatores. O primeiro deles seria o preço de custo real de uma automatização, pois os componentes utilizados no presente trabalho foram uma escolha particular, podendo ser diferentes daqueles utilizados por empresas, que podem optar por módulos mais caros ou até mesmo com melhor desempenho. Além disso, muitas empresas requerem também que reformas sejam realizadas a fim de instalarem o sistema. Todos estes fatores aumentam o preço do produto final.

Deve-se levar em consideração também que a crença popular de que sistemas automatizados são altamente tecnológicos e, conseqüentemente, caros, abre precedentes para a exploração financeira dos mesmos. Assim, este fator contribui para que a comercialização destes produtos se mantenha elitizada e com um valor pouco praticável para a maioria da população, mesmo que os valores de instalação não sejam tão elevados assim.

4 CONCLUSÃO

Este trabalho propôs um sistema de automação residencial com sensores controlados por um aplicativo móvel para Android. Desde o princípio decidiu-se por um sistema baseado em redes sem fio para oferecer maior flexibilidade ao usuário e evitar a necessidade de reformas na residência. Este último fator é muito importante, pois diminui consideravelmente o custo total do projeto.

A principal dificuldade encontrada no decorrer do trabalho foi referente à aquisição dos componentes eletrônicos, em especial dos módulos XBee e dos adaptadores USB. Devido à pouca oferta no mercado brasileiro, estes itens precisaram ser importados, revelando a necessidade de incentivos nacionais para a área. Vale ressaltar que tais incentivos seriam ainda extremamente importantes economicamente falando, uma vez que o mercado consumidor brasileiro é muito promissor no tocante à aceitação da automação residencial.

O sistema desenvolvido para este trabalho está aberto a melhorias e ampliação das funcionalidades, uma vez que a automação residencial oferece muitas outras possibilidades além das abordadas neste trabalho. O usuário pode automatizar o sistema de iluminação, climatização, entretenimento, irrigação ou ainda acrescentar outros sensores e atuadores. É possível inclusive utilizar os sensores existentes para desempenharem também outras funções. Um exemplo disso seria a utilização da informação obtida do sensor de luminosidade para ligar ou desligar as lâmpadas de um ou mais cômodos da residência.

Uma melhoria interessante seria a possibilidade de oferecer ao cliente somente os módulos que o interessam, personalizando assim as soluções. Isso não impede que o sistema completo seja oferecido, mas esta abordagem global é a já existente no mercado. Assim, personalizar a solução, além de atender melhor às necessidades do cliente, pode levar também a uma diminuição dos custos de implementação do projeto de automação residencial.

Observa-se então que apesar dos contratemplos iniciais, o projeto alcançou todos os objetivos propostos, verificando-se que é possível construir sistemas de automação residencial efetivos e com melhor custo-benefício do que aqueles atualmente disponíveis no mercado. Tal fato revela que a automação residencial, ao contrário do que atualmente se pensa, não está longe de tornar-se acessível para consumidores de poder aquisitivo mediano e tornar-se um segmento promissor no mercado consumidor.

REFERÊNCIAS

FARAHANI, S. **ZigBee Wireless Networks and Transceivers**. Burlington: Newnes, 2008.

FALUDI, R. **Building Wireless Sensor Networks**. Sebastopol: O'Reilly, 2011.

MARGOLIS, M. **Arduino Cookbook**. Sebastopol: O'Reilly, 2012.

BOLZANI, C. **Residências Inteligentes**. São Paulo: Editora Livraria da Física, 2004.

ROCHOL, J. **Comunicação de Dados**. Porto Alegre: Instituto de Informática da UFRGS: Bookman, 2012. (Série Livros Didáticos, n.22).

KINNEY, P. **ZigBee Technology: Wireless Control that Simply Works**. IEEE Communications Design Conference, outubro 2003.

CARISSIMI, A. **Redes de Computadores - Notas de aula**. Porto Alegre: Instituto de Informática da UFRGS, 2011.

ESCHNER, R. H. **Sistema de Automação Residencial Baseado em Sensores ZigBee**. 2011. 38 f. Projeto de Diplomação (Bacharelado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.

GUTIERREZ, J. **IEEE Std. 802.15.4: Enabling Pervasive Wireless Sensor Networks**. Disponível em: <<http://www.cs.berkeley.edu/~prabal/teaching/cs294-11-f05/slides/day21.pdf>>. Acesso em: junho de 2014.

ASHTON, S. **ZigBee Technology Overview**. Disponível em: <<https://docs.zigbee.org/zigbee-docs/dcn/09/docs-09-5376-00-0mwg-zigbee-technology-overview.pdf>>. Acesso em: junho de 2014.

MARSDEN, I. **Network Layer Overview**. Disponível em: <<https://docs.zigbee.org/zigbee-docs/dcn/06/docs-06-4513-00-00mg-zigbee-network-layer-technical-overview.pdf>>. Acesso em: junho de 2014.

Wireless - WLAN Introdução. Disponível em: <<http://www.ti-redes.com/wireless/wlan-intro/>>. Acesso em: maio de 2014.

Redes Wi-Fi: O padrão IEEE 802.11n. Disponível em: <<http://www.teleco.com.br/tutoriais/tutorialwifiieee/default.asp>>. Acesso em: junho de 2014.

ANEXO A CÓDIGO FONTE

O sistema de automação é composto por duas partes, divididas em hardware e software. O hardware contém os circuitos e os sensores, mostrados e explicados no capítulo 3.2 deste trabalho e que são configurados e controlados pelo software implementado.

O projeto de software foi desenvolvido em três partes:

- *Smart House Control*: Aplicativo Android que permite o controle dos sensores de maneira local. Ou seja, através da rede Wi-Fi da residência.
- Software do Controlador
- Conectividade com a Nuvem

Os códigos encontram-se no endereço:

- <https://github.com/brubetinha/home-automation-project>

e para melhor compreensão será explicada a estrutura dos arquivos disponibilizados.

Na pasta Android encontra-se o projeto do aplicativo que foi desenvolvido no Eclipse. Neste projeto, os arquivos principais são:

- **AndroidManifest.xml**: que contém informações do aplicativo que serão passadas para o sistema Android.
- **Activity_main.xml e acquirement_layout.xml**: contêm a programação dos layouts das telas. Cada arquivo corresponde a uma das telas e elas encontram-se dentro da pasta res/layout/.
- **MainActivity.java e AcquirementActivity.java**: arquivos que contêm a programação do cliente Android via sockets.

Também encontra-se disponível no diretório central um arquivo .apk que permite instalar o aplicativo no celular e testar o sistema.

Na pasta Arduino está o arquivo ArduinoServer, que possui toda a implementação do controlador. Ele possui a extensão .ino pois foi programado na IDE do Arduino, que por padrão salva todos os arquivos assim. No entanto, o código foi feito em C.

O código utiliza a biblioteca Adafruit_CC3000.h, que possui funções para a configuração e uso da rede Wi-Fi e que também permite criar um servidor simples com até quatro conexões simultâneas.

Neste mesmo arquivo, uma parte do código é referente a conectividade com a Nuvem. A linha 181 do código mostra como a informação deve ser formatada e na linha 1209 inicia a função que monta e envia a requisição HTTP para a plataforma Carriots.

Informações mais detalhadas podem ser encontradas no próprio código, que está devidamente comentado.

ANEXO B TRABALHO DE GRADUAÇÃO 1

Arquitetura para automação residencial utilizando protocolos de rede sem fio

Bruna Roberta Seewald da Silva¹, Sérgio Luis Cechin²

^{1,2}Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre – RS – Brasil

{bruna.seewald, cechin}@inf.ufrgs.br

Abstract. *It is possible to interact with many different systems such as lightning, HVAC and sound separately, but it is feasible to integrate them in order to bring greater comfort and practicality for everyday life. The concept of home automation is relatively new in the Brazilian market, even though it exists for a few years in the United States of America and Europe. Since it is a novelty in Brazil, it is accompanied by high prices, which causes it to be regarded as a luxury in this country. This article presents the benefits of home automation and how it could be implemented at low cost, using some existing technologies.*

Resumo. *Interage-se com diversos sistemas como iluminação, climatização e sonorização de forma separada, mas pode-se integrá-los de maneira a trazer maior conforto e praticidade para o dia a dia. Apesar de já existir há alguns anos nos Estados Unidos e na Europa, o conceito de automação residencial é relativamente novo no mercado brasileiro. Assim, ainda é acompanhado por altos preços, fator que o deixa ser visto ainda como um luxo. Neste trabalho serão apresentados os benefícios da automação residencial e como ela pode ser implementada a baixo custo, utilizando algumas tecnologias já existentes.*

1. Introdução a automação residencial

1.1. Definição e histórico

Com o surgimento da Internet e a redução dos custos dos equipamentos, as inovações antes restritas à escritórios e indústrias agora podem ser encontradas em residências, criando um mercado mais amplo e com diversas oportunidades. O termo *Automação Residencial* advém de *Home Automation*, utilizado no mercado norte-americano. Na Europa, utiliza-se mais comumente a palavra *Domótica*, uma junção de *domus* (em

latim, casa) com Robótica (controle automatizado), termo criado pelo jornalista francês Bruno de Latour, em 1984.

Apesar de a automação predial ter seu início na década de 80, somente a partir dos anos 2000 observou-se um atento maior à automação residencial, cujos os objetivos são proporcionar maior conforto e praticidade nas tarefas diárias. Além dessa praticidade trazida pela automação residencial, fatores como segurança, sustentabilidade e valorização do imóvel foram imprescindíveis para a difusão dos benefícios dessa tecnologia.

Entre os fatores acima citados, a redução do consumo energético é um dos que mais impulsionam o mercado da automação residencial. Com a mudança nos critérios de tarifação de energia, o controle inteligente do sistema de iluminação e do sistema de climatização, por exemplo, permitem economias nos gastos mensais, além de um consumo mais eficiente.

Para efeitos de comparação, a tabela 1.1 reúne as características principais da infra-estrutura doméstica atual, ao passo que a tabela 1.2 apresenta as características da infra-estrutura doméstica automatizada proposta e seus benefícios [Bolzani 2004].

Tabela 1.1. Situação atual da infra-estrutura doméstica

Situação atual	Consequências
instalações independentes	multiplicidade de redes e cabos
redes não compatíveis	manutenção cara e complicada, dependência do fornecedor
falta de uniformidade	impossibilidade de automatização global
equipamentos limitados	dificuldade para integrar novos serviços e interligar redes, ampliação do uso de "adaptadores", obsolescência em curto prazo

Tabela 1.2. Proposta de infra-estrutura doméstica

Situação proposta	Consequências
automação de residências	maior conforto e automatização de serviços
integração dos serviços	barateamento de equipamentos e processos
centralização de sistemas	simplificação da rede
conexão com redes externas (Internet e dedicadas)	comando remoto, utilização de conteúdo digital sob demanda
monitoramento remoto de pessoas e equipamentos	facilidade de integração de novos equipamentos e serviços, rapidez no envio de alarmes, homecare
eletrodomésticos inteligentes	acesso à informação de qualquer ponto da

	casa, diminuição do tempo de procura de avarias, economia de energia
auditoria e controle de gastos	melhoria no funcionamento de sistemas, administração da residência, constante supervisão do conjunto

É importante salientar que os desejos dos usuários sempre devem prevalecer no momento da implantação do sistema automatizado e quanto a automação vai poder auxiliar no tocante à economia e ao conforto, dependerá do tipo de vida, dos gostos pessoais e dos recursos disponíveis [Bolzani 2004].

1.2. O mercado atual

A evolução do mercado trouxe a necessidade de formação de profissionais especializados no assunto, chamados atualmente de Integradores de Sistemas Residenciais. Este profissional deve conhecer a maior gama de tecnologias e opções disponíveis para propor e realizar um projeto de maneira eficiente e que atenda às necessidades do cliente. Como estes projetos tendem a ser complexos, o profissional deve estar presente em todas as etapas, desde a concepção até os ajustes finais, trabalhando em conjunto com os profissionais de outras áreas.

Segundo [Muratori e Dal Bó 2013], o mercado atualmente se posiciona da seguinte forma:



Conforme esta distribuição, os desenvolvedores podem ser empresas de software que procuram desenvolver um protocolo de interoperabilidade entre os equipamentos e sistemas residenciais.

Os integradores também podem atuar como revendedores, já que possuem os conhecimentos técnicos necessários, além de oferecer auxílio pós-venda e manutenção dos sistemas.

É imprescindível para o sucesso do Integrador de Sistemas que ele esteja atualizado sobre as soluções existentes, pois novas técnicas surgem constantemente e elas podem ser decisivas para as necessidades do cliente. Segundo [Bolzani 2004]

também é importante trabalhar em equipe, pois com tantos profissionais envolvidos na construção ou reforma de uma residência, podem surgir entraves ou dúvidas sobre o projeto de automação residencial e será o Integrador o responsável por propor soluções para a continuidade da obra.

Por último tem-se os clientes, que são os grandes beneficiados pelas inovações contínuas do mercado tecnológico. Assim, a geração atual, acostumada com mudanças rápidas e já familiarizada com os benefícios que várias tecnologias trouxeram, aceita muito bem a automação de processos. Desse modo, há um grande interesse por parte dos consumidores em automatizar também suas casas, vista a praticidade e futura economia trazidos pelo processo. No entanto, o consumidor deve ter liberdade e controle assegurados e por isso, segundo [Bolzani 2013], a mudança em relação ao paradigma convencional de "quanto mais tecnologia melhor" sugere o desenvolvimento de equipamentos e serviços com foco no usuário e não na tecnologia em si. O objetivo é aumentar os benefícios que os sistemas eletrônicos podem proporcionar às pessoas, especialmente no ambiente doméstico, e com isso desmistificar a automação residencial e o conceito de casas inteligentes, bem como estimular o desenvolvimento, a produção e o mercado.

2. Os principais sistemas residenciais

Nas residências existem muitos sistemas denominados autônomos, que operam de forma independente, exercendo todas as funções para as quais foram projetados, sem a necessidade da intervenção de um controlador centralizado [Muratori e Dal Bó 2013].

A seguir detalham-se os sistemas que podem ser automatizados e as suas principais partes constituintes.

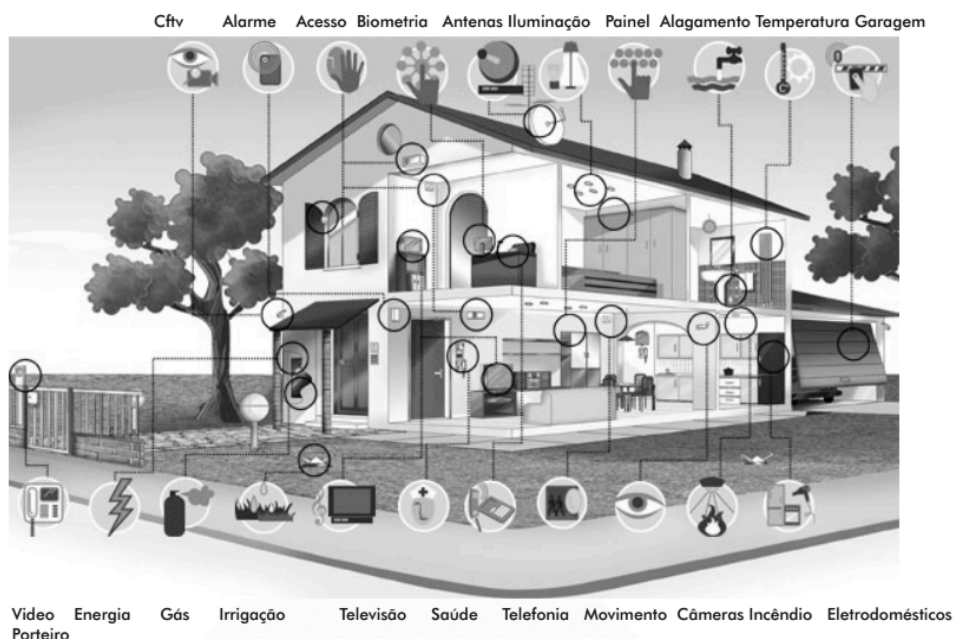


Figura 1. Sistemas Residenciais¹

¹Imagem retirada de: <http://www.knxdobrasil.com.br/folders/Catalogo-Knx-05.pdf>. Acesso em: 20/03/2014.

2.1. Iluminação

Historicamente, os equipamentos de controle de iluminação sempre foram o ponto de partida no processo de automação de uma casa [Bolzani 2004]. Segundo [Muratori e Dal Bó 2013] pode-se dividir o sistema de iluminação em duas partes: elétrico e natural.

O sistema elétrico é constituído por todas as lâmpadas alimentadas por energia elétrica. É possível por meio de um processo conhecido como dimerização variar-se gradativamente a intensidade luminosa dessas lâmpadas, de modo a criar diversos ambientes, sempre de acordo com o gosto do cliente. Para tal, as lâmpadas podem ser dimerizadas por controladores que utilizam componentes eletrônicos específicos para cada tipo e são chamados de módulos dimmer.

Já o sistema de iluminação natural consiste em dispositivos motorizados que abrem e fecham, permitindo ou bloqueando assim a passagem de luz natural. Estes dispositivos também são acionados por controladores que utilizam saídas a relés, usualmente conhecidas como saídas liga/desliga.

São exemplos de dispositivos motorizados:

- Cortinas com trilhos motorizados
- Venezianas
- Persianas
- Painéis
- Coberturas motorizadas

Conforme [Bolzani 2004] com a automação do sistema de iluminação, pode-se acender ou apagar luzes em horários definidos, programá-las conforme a estação do ano e/ou ambiente ou determinar qual será a intensidade luminosa. Isto em conjunto com sensores de presença e temporizadores reforça a idéia do consumo eficiente de energia.

2.2. Climatização

O sistema de climatização é composto principalmente por equipamentos de ar condicionado, ventiladores, aquecedores e pisos aquecidos. Todos estes equipamentos já possuem algum tipo de controle remoto infravermelho incorporado, portanto a integração com o sistema automatizado pode ser feita utilizando emissores de infravermelho. No caso dos ventiladores, ainda existem os controladores locais, que ficam próximos ao sistema de iluminação do ambiente. No caso de sistemas para aquecimento do ambiente, por questões de segurança, dá-se preferência para equipamentos que utilizem a energia elétrica [Muratori e Dal Bó 2013].

Para aumentar a eficiência e preparar a residência de acordo com o gosto do cliente, tanto o sistema quanto o cliente devem ter acesso aos dados enviados pelos sensores de temperatura, umidade, ventilação de cada ambiente. Assim, é possível verificar a necessidade de ligar/desligar algum equipamento, mesmo que remotamente.

2.3. Áudio e Vídeo

As residências modernas possuem muitos equipamentos de áudio e vídeo tais como televisores, players de DVD/Bluray, consoles de videogames e decoders de TV a cabo. Segundo [Muratori e Dal Bó 2013] com tantas opções, é evidente a importância deste sistema, ainda mais que ele se caracteriza por ser uma fonte de entretenimento para o cliente.

Ainda conforme [Muratori e Dal Bó 2013], na parte de sonorização ambiente, é comum existir um amplificador central composto por várias zonas de sonorização para possibilitar que diferentes ambientes possam tocar músicas distintas. Assim, enquanto alguém assiste a um filme na sala, é possível escutar música em outro cômodo, sem a necessidade de aparelhagem extra. Como este sistema está atrelado ao fator estético da residência, uma grande variedade de formatos para as caixas acústicas que serão instaladas junto ao sistema de sonorização podem ser utilizados.

2.4. Sensores e atuadores

Em automação os sensores são utilizados para monitorar e fornecer informações sobre os ambientes da residência. Estes sensores são dispositivos que transformam grandezas físicas, como temperatura, umidade e pressão, em um sinal elétrico correspondente. Já os atuadores realizam a conversão do sinal elétrico em energia mecânica e produzem uma ação.

A seguir são dados alguns exemplos, para melhor compreender essa relação e como elas podem existir nas residências [Muratori e Dal Bó 2013]:

- Caso o detector de gás seja ativado, o sistema fechará a válvula solenóide instalada na entrada de gás da residência, inibindo assim o vazamento de gás;
- Caso o detector de inundação seja ativado, o sistema fechará a válvula solenóide instalada na entrada de água, inibindo assim o vazamento de água;

Além disso, pode-se programar o sistema para que o usuário seja alertado da ocorrência de algum destes casos.

No mercado existem diversos tipos de sensores que podem ser integrados ao sistema de automação para auxiliar no monitoramento e controle dos ambientes, alguns exemplos são:

- Qualidade do ar
- Magnéticos de abertura
- Intensidade luminosa
- Fumo
- Gás
- Movimento
- Chama
- Chuva e umidade do solo

2.5. Irrigação

Este sistema envolve jardins, gramados, vasos de plantas e sua lógica de funcionamento é simples. A partir dos dados coletados por alguns sensores, programa-se para que a irrigação seja feita em horários pré-determinados, caso necessário. Por exemplo, se os sensores detectarem chuva ou que a umidade do solo está muito alta, o jardim não será irrigado, evitando assim o desperdício de água.

No caso de não haver um sistema com aspersores e rotores que irrigue as plantas automaticamente, pode-se alertar o usuário quando ele deve realizar esta ação. Isto pode ser feito por meio de mensagens de texto pelo celular ou emails, por exemplo.

2.6. Integração com sistemas de segurança

Os sistemas de segurança são os mais utilizados nas residências. Eles possuem grande complexidade e desempenham suas funções através de um hardware dedicado. Este sistema ainda pode ser dividido em subsistemas segundo [Bolzani 2004]:

- Detecção perimetral: detecção de intrusos através de centrais de choque com hastes e arames ou sensores infravermelhos.
- Sensoriamento interno: composto por sensores magnéticos, de presença, quebra de vidro, enviando sinais no caso de alguma anormalidade.
- Circuito fechado de televisão: composto por um conjunto de câmeras instaladas em pontos críticos para supervisão dos locais e coibir a ação de criminosos.
- Controle de acesso: permissão ou não da entrada de pessoas, por meio de um dispositivo de identificação, por exemplo, biométrico.
- Controle de rondas: feito pelos seguranças, caso hajam.

A integração com o sistema de automação serve para aumentar a eficácia do sistema de segurança e geralmente são utilizados os sinais de *status* da central de alarme [Muratori e Dal Bó 2013]. Assim, é possível armar/desarmar remotamente os alarmes e receber notificações caso eles tenham disparado.

3. Principais protocolos de automação

3.1. Protocolos de automação *powerline*

Redes de transmissão de dados baseadas no mesmo meio físico da distribuição de energia elétrica são conhecidas como redes *powerline*. Elas permitem que se utilize a fiação elétrica existente na residência como meio de transporte de pacotes de dados [Bolzani 2004].

Porém muitos problemas impediram o progresso desta tecnologia, entre eles incidência de ruídos, atenuação do sinal e variações dos níveis de impedância.

3.1.1 X10

O padrão surgiu na década de 70 e é considerado um dos primeiros desenvolvidos para automação residencial. A patente original expirou em 1997, permitindo que novos produtos baseados em X10 fossem lançados no mercado.

O protocolo funciona da seguinte maneira: as mensagens são formadas por 13 bits, sendo os 4 primeiros o *Start Code*, os 4 seguintes o *House Code* (de A a P), os próximos 4 identificam uma função (ligar, desligar, etc) ou um número (de 1 a 16) e o último bit determina qual destas duas opções, função ou número, será utilizada. Portanto, este protocolo permite o controle de 256 equipamentos e possui 16 comandos, que são limitados a ligar, desligar ou alterar a intensidade.

Para enviar um comando são necessárias pelo menos duas mensagens, uma com o endereço e a outra com o comando em si. Além disso, para assegurar o recebimento das mensagens, elas são enviadas duas vezes, em razão à existência de ruído na rede.

Devido ao número limitado de funções e a baixa integração com outros sistemas, este protocolo não é o mais indicado para automação, recorrendo-se a outras soluções.

3.1.2 Home Plug Powerline Alliance

Esta aliança surgiu no início dos anos 2000 e visa criar especificações utilizando a rede elétrica. Vários padrões foram lançados desde então, começando com o Home Plug 1.0, de 2001.

O Home Plug AV surgiu em 2005 com a promessa de atingir grandes velocidades ao trabalhar com vídeo, áudio e dados. As principais mudanças ocorreram na camada física, na subcamada MAC - que foi projetada para ser altamente eficiente, suportando TDMA e CSMA -, e na segurança, utilizando uma criptografia AES-128 bits.

O Home Plug AV2 surgiu em 2012 e é compatível com o padrão IEEE 1901. Este padrão procura atender à necessidade de altas velocidades na integração dos diversos serviços presentes nas residências modernas.

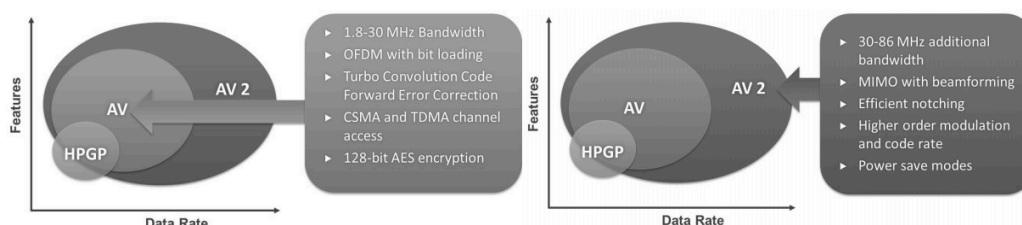


Figura 2. Diferenças entre os padrões AV e AV2²

3.2. Protocolos de automação sem fio

3.2.1. IEEE 802.11

A família de padrões IEEE 802.11 define as estruturas básicas para a construção de uma rede local sem fio (WLAN). A estrutura básica do protocolo define políticas de controle e de acesso ao meio físico, ao ar e corresponde às camadas física e de enlace do modelo OSI [Dau 2013].

O IEEE 802.11 é voltado para a comunicação entre computadores por possuir um alcance maior (chegando à algumas centenas de metros) em substituição às redes LANs cabeadas. Ao longo do tempo diversas versões surgiram, com modificações em termos de frequência de operação, largura de banda e alcance da rede.

Com a popularização das redes sem fio, em especial do Wi-Fi, a maioria dos produtos possuem suporte para esta tecnologia, tornando-a um candidato a ser utilizado na arquitetura futuramente proposta para a automação residencial.

3.2.2. Bluetooth

O Bluetooth também é um padrão com especificações para a camada física e de enlace mantido pelo Bluetooth SIG. É destinada para comunicações de curto alcance, conectando joysticks, mouses, teclados, fones de ouvido, por exemplo e formando as redes pessoais sem fio (WPANs). O Bluetooth opera na faixa de 2.4GHz, a mesma que outros padrões utilizam e as velocidades de transmissão [Dau 2013] podem chegar a 24Mbits/s na versão 3.0. No entanto, a taxa de transmissão mais comum é a de até

²Imagem retirada de Home Plug AV2 White Paper, p.6-7.

3Mbits/s, definida na versão 2.0.

A topologia mínima é denominada *piconet*, onde os dispositivos se conectam na forma mestre-escravo, com um mestre e no máximo sete escravos.

Destinado aos dispositivos móveis, em que o consumo de energia é um fator extremamente importante, o Bluetooth é uma boa alternativa. No entanto, outras soluções foram desenvolvidas para trabalhar diretamente com sensores e redes residenciais e algumas delas estão apresentadas abaixo.

3.2.3. ZigBee

O ZigBee é uma tecnologia desenvolvida pela ZigBee Alliance, voltada para redes de sensores e dispositivos de controle. O objetivo principal do protocolo é consumir pouca energia, fator que possibilita vida longa às baterias dos dispositivos móveis e facilita a manutenção.

As camadas física e de enlace são baseadas no padrão IEEE 802.15.4, operando na faixa 2.4GHz para todo o mundo. Nesta faixa as velocidades de transmissão são de 250 kbits/s.

Em termos de comparação com o Bluetooth, o ZigBee também necessita de um dispositivo que coordene e mantenha a rede. Porém ele permite um número muito maior de dispositivos conectados, quando comparado com os oito do Bluetooth.

Estudos mostram que estes dois protocolos não são concorrentes, mas sim soluções para áreas diferentes. Enquanto o ZigBee foi estruturado para ser eficiente em redes de sensores e por isso o consumo energético é muito menor, o Bluetooth é a melhor opção para dispositivos que necessitam de maiores taxas de transmissão. Visto que o projeto propõe baratear custos e possuirá módulos que serão alimentados por baterias e constituídos por sensores, o protocolo ZigBee é um candidato para ser utilizado na arquitetura que será proposta.

3.2.4. Z-Wave

O Z-Wave é um protocolo proprietário, desenvolvido pela Sigma Designs, destinado à automação residencial. O principal objetivo deste protocolo é permitir transmissões confiáveis de uma unidade central para outros nodos da rede, de forma semelhante aos protocolos anteriores. Ele opera na faixa em torno dos 900MHz e permite transmissões de 9.6 kbits/s e 40 kbits/s.

A grande vantagem deste protocolo é que ele foi desenvolvido para ser o padrão em soluções de automação residencial, no entanto, esta solução não é aberta, tornando a utilização deste inviável neste trabalho.

3.3. Protocolos de automação híbridos

3.3.1. LonWorks

A Echelon Corporation desenvolveu na década de 80 um sistema completo de automação voltado para o mercado predial e industrial. Está presente hoje em aproximadamente 100 milhões de equipamentos e oferece diversos produtos divididos em transceivers, controladores e ferramentas de software. O protocolo LonWorks é aberto e tornou-se com o passar do tempo um padrão ANSI, IEC e, mais recentemente,

ISO.

A pilha de protocolos é complexa e usa as setes camadas do modelo OSI. Na camada física são suportadas conexões cabeadas, sem fio, por fibra ótica, powerline, por exemplo.

3.3.2. KNX

Este padrão surgiu no final da década de 90, resultante da união de três associações: EIB (*European Installation Bus*), EHS (*European Home System*) e BatiBUS.

O KNX é uma norma aberta e internacional (ISO/IEC 14543-3) dedicada ao mercado residencial e predial. Além disso é independente de hardware ou software e os fabricantes que desejarem certificar seus produtos devem estar de acordo com a ISO 9001. Até o momento os meios físicos padronizados são: cabos de pares trançados, powerline, radiofrequência e ethernet/IP.

Assim como o LonWorks, o KNX ainda é mais utilizado em sistemas comerciais e apesar de fornecer uma solução completa para a automação, sua aplicação prática é bastante reduzida. As soluções utilizando tecnologias mais conhecidas tem tido preferência nas instalações de residências inteligentes.

4. Arquitetura para automação residencial sem fio

4.1. O projeto

A proposta deste trabalho é possibilitar a automação de uma residência utilizando a rede sem fio. O objetivo é desenvolver uma arquitetura que faça a integração entre dispositivos móveis (tablets, smartphones, por exemplo) e os equipamentos controlados.

O projeto visa facilitar a implantação, diminuir os custos e expandir a automação residencial no mercado brasileiro, que se encontra atualmente limitada, oferecendo opções geralmente caras.

A decisão sobre os componentes que farão parte da arquitetura foi baseada principalmente no protocolo de comunicação. Após estudar as diferentes opções de protocolos sem fio optou-se por utilizar Wi-Fi e ZigBee, sendo o primeiro escolhido pela enorme difusão nos dispositivos móveis e nas próprias residências (o que reduz os custos com uma parte da estrutura) e o segundo pelas suas características de eficiência energética. Sendo assim, verifica-se a necessidade de um controlador que fará a integração entre esses dois protocolos.

O projeto será composto por duas partes: a primeira parte será a implementação do aplicativo para Android e a segunda parte será a implementação do controlador utilizando uma placa da plataforma Arduino. As explicações e detalhes sobre o hardware serão dadas no capítulo 4.2 e sobre o software no capítulo 4.3.

4.2. O hardware

A arquitetura será composta principalmente por um controlador, que realizará a integração entre os protocolos Wi-Fi e ZigBee. Pela facilidade de uso e pela fartura de documentação, foi escolhida uma placa da plataforma Arduino.

4.2.1. A placa Arduino

O Arduino Mega 2560 é uma placa microcontroladora baseada no ATmega 2560, com 54 entradas/saídas digitais (das quais 15 podem ser saídas PWM), 16 entradas analógicas, 4 UARTs, um cristal de 16 MHz, conexão USB, conector para alimentação externa, memória flash de 256 KB, 8 KB de SRAM e 4 KB de EEPROM.

Junto à placa serão utilizados *Shields WiFi* e *XBee*, que servem para ampliar as capacidades das placas Arduino. Neste caso, as opções de comunicação via Wi-Fi e módulos XBee são integradas ao Arduino.

Os módulos de rádio XBee *Series 2* são fabricados pela Digi International e suportam uma variedade de protocolos de comunicação, entre eles o ZigBee. Estes módulos serão responsáveis pela comunicação entre o controlador e os equipamentos espalhados pela residência. Já o módulo Wi-Fi será responsável pela comunicação entre os dispositivos móveis e o controlador. A figura 3 mostra a placa Arduino Mega e dois modelos de rádio XBee e a figura 4 mostra um exemplo da arquitetura proposta.

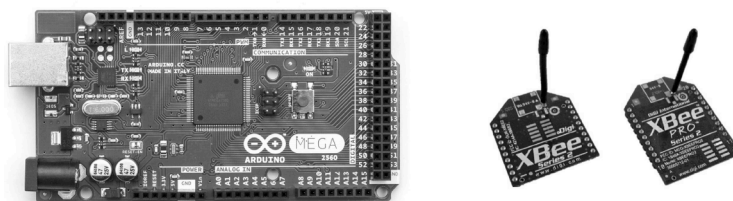


Figura 3. Arduino Mega e módulos Xbee³

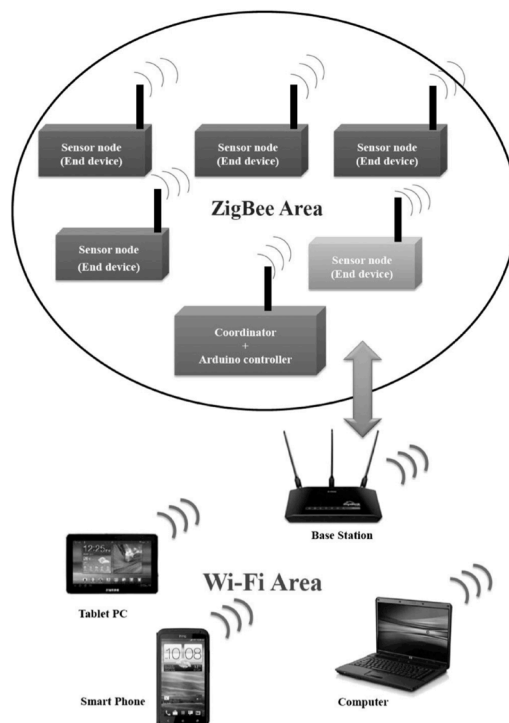


Figura 4. Arquitetura proposta⁴

³Imagem retirada de Lian et al., 2013. p.758 e <http://arduino.cc/en/Main/ArduinoBoardMega2560>. Acesso em: 22/03/2014.

4.3. O software

A implementação deste trabalho será realizada em dispositivos móveis do tipo *smartphone*, com sistema operacional Android. Será utilizada a API 16, lançada na versão *Jelly Bean* (4.1). O desenvolvimento do código fonte será na linguagem Java, padrão para desenvolvimento Android e na linguagem XML, padrão para definição gráfica no Android. O aplicativo permitirá o controle dos recursos disponíveis a partir do *smartphone*.

4.3.1. O aplicativo

Para este trabalho foram escolhidos os sensores de temperatura, umidade, luminosidade, movimento e gás. Será possível verificar através do aplicativo as informações dos sensores e ainda poderão ser executadas as ações de (des)acionamento dos sensores e emissão de sinais sonoros para indicar anormalidades detectadas por algum dos sensores.

Para a comunicação entre o aplicativo e o controlador optou-se pelo modelo cliente-servidor, utilizando sockets TCP/IP. No aplicativo Android será implementado o cliente enquanto no controlador será implementado o servidor.

4.3.2. O controlador

O controlador será implementado utilizando a linguagem de programação Arduino, padrão para desenvolvimento em placas da plataforma Arduino. A partir dele será possível conectar-se aos módulos XBee ou Wi-Fi para recebimento/transmissão de dados. O programa será também responsável pela tradução das mensagens, devido à proposta da arquitetura de trabalhar com dois protocolos. Para auxiliar a comunicação com o hardware e a montagem dos pacotes enviados pela rede serão utilizadas APIs de programação, como por exemplo a WiFi library e a Xbee library.

4.3.3. Integração com a Nuvem

Até o momento, este cenário considera somente a interação do usuário com os equipamentos de forma local, ou seja, utilizando a rede interna da residência. Porém é essencial oferecer ao usuário a opção de monitorar a sua casa à distância. Para isto será utilizada uma plataforma online, projetada para a *Internet das Coisas*, na qual é possível conectar os equipamentos da casa automatizada com a internet, permitindo o monitoramento em tempo real através de qualquer dispositivo. No caso deste trabalho, os sensores irão disponibilizar as informações para a Nuvem e na ocorrência de alguma anormalidade, o usuário será imediatamente informado por intermédio de mensagens de texto ou email.

5. Cronograma

As atividades da segunda etapa deste trabalho estão definidas conforme o cronograma abaixo:

⁴Imagem retirada de Lian et al. 2013, p.760.

Tarefa/Mês		Abril				Maio				Junho				Julho			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Configuração do ambiente de desenvolvimento	x															
2	Implementação da aplicação do usuário	x	x	x													
3	Implementação do controlador			x	x	x											
4	Integração com a Nuvem					x	x										
5	Testes da arquitetura					x	x	x	x								
6	Aquisição de dados para análise de desempenho da arquitetura									x	x						
7	Escrita da monografia													x	x	x	x

A primeira tarefa, conforme o cronograma, consiste em preparar a máquina a ser utilizada para o desenvolvimento deste trabalho. Serão instalados e configurados os ambientes para desenvolvimento no Android e no Arduino, e o terminal necessário para configuração dos módulos XBee.

Em seguida inicia-se o desenvolvimento do aplicativo a ser instalado nos dispositivos móveis do tipo *smartphone*, sucedida pela implementação do controlador e pela implementação do serviço que integra os equipamentos à internet.

Durante o mês de maio estão programados os testes com a arquitetura e o início da aquisição de dados para análise de desempenho. O mês de junho e início de julho destinam-se à escrita da monografia. Caso hajam eventuais atrasos, eles podem ser resolvidos ainda em junho.

Referências

- Bolzani, C. (2004), *Residência Inteligentes*, Livraria da Física, 1º edição.
- Muratori, J. R. e Dal Bó, P. H. (2013), *Automação Residencial Conceitos e Aplicações*, Educere, 1º edição.
- Dau, M. L. (2013), "Comunicação entre MANETs com protocolo de roteamento baseado na fonte", Instituto de Informática, Universidade Federal do Rio Grande do Sul.
- Bolzani, C., "Domótica, a nova ciência do século XXI", *Revista Fonte*, a. 10, n. 13, dezembro 2013.
- Lian, K.-Y., Hsiao, S.-J., Sung, W.-T., "Intelligent multi-sensor control system based on innovative technology integration via ZigBee and Wi-Fi networks", *Jornal of Network and Computer Applications*, v. 36(2), p. 756-767, março 2013.
- Home Plug AV White Paper, disponível em http://www.homeplug.org/tech/whitepapers/HPAV-White-Paper_050818.pdf, Março
- Home Plug AV2 White Paper, disponível em http://www.homeplug.org/tech/whitepapers/HomePlug_AV2_whitepaper_130909.pdf, Março
- The LonWorks® Protocol, <https://www.echelon.com/technology/lonworks/lonworks-protocol.htm>, Março

KNX Association, <http://www.knx.org/knx-en/index.php>, Março
Arduino, <http://arduino.cc/>, Fevereiro