

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**Marcelo Peres Polese**

**MONITORAÇÃO E ATUAÇÃO VIA WEB  
BASEADO EM REDES ZIGBEE**

Porto Alegre  
Julho de 2013

MARCELO PERES POLESE

**MONITORAÇÃO E ATUAÇÃO VIA WEB  
BASEADO EM REDES ZIGBEE**

Trabalho de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Marcelo Götz

Porto Alegre

2013

MARCELO PERES POLESE

**MONITORAÇÃO E ATUAÇÃO VIA WEB  
BASEADO EM REDES ZIGBEE**

Este Trabalho de Diplomação foi julgado adequado como pré-requisito para a obtenção do título de ENGENHEIRO ELETRICISTA e aprovado em sua forma final pelo Professor Orientador e pelo Coordenador da disciplina Projeto de Diplomação Engenharia Elétrica (ENG04029) da Universidade Federal do Rio Grande do Sul.

---

Prof. Dr. Marcelo Götz  
Orientador

---

Prof. Dr. Altamiro Amadeu Susin  
Coordenador

Aprovado em: \_\_\_/\_\_\_/\_\_\_\_\_

BANCA EXAMINADORA

Prof. Dr. Marcelo Götz – UFRGS

\_\_\_\_\_

Prof. Dr. Carlos Eduardo Pereira – UFRGS

\_\_\_\_\_

Prof. Dr. Flávio Antônio Becon Lemos – UFRGS

\_\_\_\_\_

Dedico este trabalho a meus pais, Deoclécio Marcos Polese e Vanilde de Bovi Peres, que sempre me apoiaram e, especialmente durante o período do meu Curso de Graduação, estiveram ao meu lado.

## **AGRADECIMENTOS**

Agradeço ao meu professor orientador, Marcelo Götz, pelo apoio e paciência durante a realização deste trabalho.

Aos meus pais, que além de me ensinarem o valor do estudo, me ensinaram o valor do caráter e da retidão.

A minha namorada, companheira de todas as horas, pelo apoio incondicional às minhas decisões.

Aos meus amigos, pelos momentos de alegria e descontração.

À UFRGS e a École Centrale de Lyon, que me proporcionaram educação de qualidade através de excelentes mestres.

Existe uma coisa que uma longa existência me ensinou: toda a nossa ciência, comparada à realidade, é primitiva e inocente; e, portanto, é o que temos de mais valioso.

*Albert Einstein*

## RESUMO

Este trabalho é um estudo sobre o tema Automação Residencial. Em um primeiro momento, uma análise sobre o conceito de Automação Residencial é apresentada. Em uma segunda etapa, um estudo comparativo entre as diferentes tecnologias existentes é desenvolvido, apresentando seus respectivos pontos fortes e fracos. Por último, o sistema desenvolvido, baseado no protocolo de comunicação *wireless* Zigbee, é descrito tecnicamente em detalhe. Os resultados obtidos permitem a criação de um sistema de atuação/monitoramento com interface *Web* de baixo custo.

**Palavras-chave:** Automação Residencial. Zigbee. Gateway. Atuação Remota.

## ABSTRACT

This work is a study on the topic of Home Automation. At first, an analysis of the concept of Home Automation is presented. In a second stage, a comparative study between the different existing technologies is developed, presenting their respective strengths and weaknesses. Finally, the developed system based on Zigbee wireless communication protocol is technically described in detail. The results obtained allow the creation of a low cost system for home monitoring/actuation provided with web interface.

**Keywords:** Home Automation. Zigbee. Gateway. Remote actuation.



## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	09
1.1 MOTIVAÇÃO E METODOLOGIA .....	10
1.2 ESTRUTURAS DO TRABALHO .....	11
<b>2 AUTOMAÇÃO RESIDENCIAL: CONCEITO, MERCADO E TECNOLOGIAS</b> .....	12
2.1 CONCEITO .....	12
2.2 SCHNEIDER ELECTRIC.....	14
<b>2.2.1 Linha IHC- Intelligent Home Control</b> .....	14
<b>2.2.2 Liha BUS KNX</b> .....	16
2.3 ABB .....	18
2.4 SIEMENS .....	19
2.5 OUTROS SISTEMAS E PROTOCOLOS .....	20
<b>3 SISTEMA DESENVOLVIDO: XHOME</b> .....	23
3.1 FUNCIONALIDADES E ARQUITETURA .....	23
3.2 REDES XBEE: CONCEITO E APLICAÇÃO.....	25
<b>3.2.1 Características técnicas de redes Xbee</b> .....	26
<b>3.2.2 Modos de operação do Xbee</b> .....	29
<b>3.2.3 Configurações dos dispositivos na rede Xhome</b> .....	32
3.3 GATEWAY ETHERNET/ZIGBEE .....	34
<b>3.3.1 Arduino Uno</b> .....	34
<b>3.3.2 Shield Ethernet para Arduino</b> .....	36
<b>3.3.3 Placa de interface Arduino/Xbee</b> .....	36
<b>3.3.4 Programação do microcontrolador</b> .....	38
3.4 ATUADORES E SENSORES.....	40
3.5 XIVELY.....	42
3.6 ANALISE DA PERFORMANCE DO SISTEMA .....	43
<b>3.6.1 Análise do tempo de resposta do sistema</b> .....	44
3.6.1.1 Análise do tempo de atualização dos atuadores .....	45
3.6.1.2 Análise do tempo de atualização dos sensores .....	47
<b>3.6.2 Outros parâmetros</b> .....	48
<b>4 CONCLUSÃO</b> .....	50
REFERÊNCIAS.....	52
ANEXO A .....	54

## 1 INTRODUÇÃO

O mercado de automação residencial é um mercado relativamente novo e em plena expansão. O seu desenvolvimento segue a tendência de automação dos processos que foi visto na indústria há alguns anos atrás. Em outras palavras, a mesma revolução que as tecnologias de automação propiciaram na indústria deve ser visto no meio residencial/predial nos próximos anos.

Apesar dos meios tecnológicos para o desenvolvimento destes sistemas já serem dominados há algum tempo (ex.: eletrônica embarcada necessária), a sua expansão tem se dado de maneira mais lenta e retardada em relação à automação industrial. Este fenômeno tem várias explicações, porém dois fatores básicos para entendê-lo são “custo” e “benefícios”.

Por se tratarem de dois mercados diferentes, um composto por grandes empresas e um volume financeiro elevado e outro formado por consumidores residências, cujo poder aquisitivo é muito menor, é natural que o desenvolvimento aconteça de maneira mais lenta e gradual neste segundo mercado. Muitas vezes o investimento inicial necessário para implantação de um sistema de automação residencial inviabiliza financeiramente o mesmo ou ao menos o restringe a um público residencial muito seletivo, com alto poder aquisitivo.

Além do fator custo, os benefícios que entram em jogo são diferentes. Ao nível industrial, a automação traz benefícios como aumento no volume de produção e diminuição do custo de produção unitário. Uma rápida análise financeira pode demonstrar que o investimento inicial desembolsado para implantação do sistema de automação se pagará em alguns anos. Desta forma, os sistemas de automação industrial tornam as empresas mais competitivas e são meios importantes, quiçá essenciais, para que as mesmas sobrevivam no mercado.

Os benefícios dos sistemas de automação ao nível residencial/predial são mais subjetivos e muitas vezes complicados de justificar financeiramente. Entre as principais razões que levam um usuário a optar pela implantação de um sistema de automação residencial são o conforto, segurança, praticidade e eficiência energética. Dentre estes quatro benefícios apresentados, o único que se justifica financeiramente de uma maneira mais fácil é a eficiência energética.

Por uma série de razões, porém, este cenário vem mudando e o mercado de automação residencial está crescendo aceleradamente. O principal motivo por traz

dessas mudanças é a produção em massa de dispositivos microprocessados, além de melhorias no processo de produção desses sistemas. Atualmente, os sistemas embarcados estão presentes cotidianamente em nossas vidas, dos *tablets* e *netbooks* aos sistemas de controle de injeção de combustíveis em automóveis. Esta generalização da utilização dos sistemas embarcados reflete diretamente na diminuição do preço dos mesmos, impactando, entre outros, os sistemas de automação. Em outras palavras, os sistemas de automação residencial têm se tornado viáveis financeiramente para uma camada cada vez maior da sociedade. Com efeito, o sistema proposto neste trabalho demonstra a viabilidade de se construir um sistema de automação residencial comandado remotamente com baixo orçamento.

Além da questão financeira, por uma mudança nos valores atuais da sociedade, questões como segurança e conforto têm se tornado cada vez mais relevantes para os usuários. Em decorrência do direcionamento positivo destes aspectos cruciais (benefícios e preço), o mercado de automação residencial vem crescendo fortemente. De fato, estimativas apontam que o setor faturou 4 bilhões de reais em 2012 e deve crescer mais 30% em 2013 (G1 O portal de notícias da Globo, 2013).

## 1.1 MOTIVAÇÃO E METODOLOGIA

Um sistema maduro de automação residencial demanda conhecimentos em diversas áreas: protocolos de comunicação, programação de microcontroladores, microeletrônica, controle, softwares, instalações elétricas residenciais, entre outras. Essa multidisciplinariedade é sem dúvida um fator desafiante e motivador para os engenheiros que aspiram trabalhar no setor. Somando-se às questões técnicas, o mercado, conforme exposto posteriormente, apresenta tendências favoráveis de crescimento. Estes e outros aspectos foram de fundamental importância para a escolha do tema de trabalho.

Este trabalho tem por objetivo desenvolver um sistema de automação residencial que possua uma interface com a internet. Deve ser possível monitorar e acionar dispositivos localizados dentro da residência remotamente. Por fim, este sistema deve ser preferencialmente de baixo custo.

A metodologia adotada para desenvolvimento do sistema segue os padrões

clássicos do modelo em V de desenvolvimento em Engenharia de Sistemas (Modelo V, 2012). Inicialmente foi feita uma análise conceitual/operacional do que se esperava do sistema, de onde foram derivados os requisitos funcionais que o mesmo deveria atender. Com base nos requisitos funcionais, uma arquitetura foi proposta e após detalhada. Para implementar a arquitetura, alguns hardwares de mercado foram selecionados, ao passo que outros foram desenvolvidos. Por último, testes de integração permitiram validar o sistema. O método utilizado será explicitado com maiores detalhes no capítulo 3, referente ao sistema desenvolvido.

## 1.2 ESTRUTURA DO TRABALHO

Para melhor entendimento do tema estudado, o trabalho foi dividido em 2 partes: “Automação residencial: Conceito, Mercado e Tecnologias” e “Sistema Desenvolvido: Xhome”.

O capítulo 2, referido como “Automação residencial: Conceito, Mercado e Tecnologias”, visa, em um primeiro momento, definir o que é “automação residencial” e quais são os principais benefícios que, de maneira qualitativa, um usuário pode auferir ao utilizar-se tal sistema. Em seguida, uma análise técnica dos sistemas de automação residencial existentes atualmente é apresentada. Tanto as empresas atuantes, quanto as soluções propostas são revisadas tecnicamente. A gama de soluções abrange, entre outras, sistemas cabeados (diversos protocolos), utilização da rede elétrica para comunicação (tecnologia PLC) e tecnologias Wireless.

No capítulo 3, a solução desenvolvida é apresentada com detalhamento técnico. As justificativas para a escolha da topologia e a metodologia de desenvolvimento também são explicitadas. Adicionalmente, uma análise da performance do sistema será feita.

Por fim, na conclusão do trabalho, um contraste entre a solução desenvolvida e os sistemas de mercado é apresentado e são sugeridos pontos para continuação e melhoria deste trabalho.

## 2 AUTOMAÇÃO RESIDENCIAL

### 2.1 CONCEITO

O termo automação residencial está cada vez mais difundido na mídia, sendo às vezes utilizado de maneira equivocada ou errônea. Outro termo que busca designar o mesmo conceito é a palavra “domótica”, junção de “domus” (casa) e “robótica”. Na realidade, não existe uma definição formal e precisa do termo, apenas definições similares, provenientes de diferentes entidades, e que visam explicitar o mesmo conceito.

Na Europa, o termo mais difundido é domótica. Ao passo que no Brasil, a tendência é optar por automação residencial, tradução do termo empregado nos Estados Unidos – *home automation*.

Uma definição adequada e explicativa é dada pela CEDOM (asociación española de domótica) em (ROBERTO MURATORI, José; HENRIQUE DAL BO, Paulo, 2012):

*é a automatização e o controle aplicados à residência. Esta automatização e controle se realizam mediante o uso de equipamentos que dispõem de capacidade para se comunicar interativamente entre eles e com capacidade de seguir as instruções de um programa previamente estabelecido pelo usuário da residência e com possibilidades de alterações conforme seus interesses. Em consequência, a domótica permite maior qualidade de vida, reduz o trabalho doméstico, aumenta o bem-estar e a segurança, racionaliza o consumo de energia e, além disso, sua evolução permite oferecer continuamente novas aplicações.*

Alguns termos importantes são utilizados nesta definição, e que servem para melhor entender o conceito geral. Entre eles, os conceitos de controle e automação, já bem instituídos e conhecidos em engenharia elétrica e a noção de interação entre equipamentos. Em outras palavras, o sistema de automação residencial permite integrar as diversas soluções tecnológicas pontuais em um sistema maior, auferindo-se benefícios desta integração.

Exemplificando, um sistema de controle de persianas acionado remotamente ou um sistema de *Home Theater* não podem ser considerados sistemas de automação residencial, apenas soluções tecnológicas isoladas. A partir do momento em que estes sistemas são dotados da capacidade de comunicar, permitindo, por

exemplo, de maneira integrada fechar as persianas e ligar o sistema de *DVD/Home Theater* para preparar um cenário para projeção de filmes, estes podem ser considerados elementos de um sistema de automação residencial.

Entre os diversos sistemas de uma casa que podem ser integrados em um sistema residencial único, pode-se citar:

- *Instalação elétrica*, que compreende: iluminação, tomadas, gestão de energia e outros;
- *Sistema de segurança*: alarmes de intrusão, alarmes técnicos (fumaça, vazamento de gás, inundação), circuito fechado de TV, monitoramento, controle de acesso;
- *Sistemas multimídia*: áudio e vídeo, som ambiente, jogos eletrônicos, além de vídeos, imagens e sons sob demanda;
- *Sistemas de comunicações*: telefonia e interfonia, redes domésticas, TV por assinatura;
- *Utilidades*: irrigação, aspiração central, climatização, aquecimento de água, bombas, persianas elétricas e outros.

A integração destes sub-sistemas proporciona ganhos em diversos aspectos. Entre eles, mas não exaustivamente, pode-se citar segurança, conforto, economia de tempo, economia de energia elétrica e praticidade.

Um exemplo de conforto e praticidade é o fato de poder controlar, de maneira integrada, a televisão e sistema de *Home Theater* através do celular ou *tablet*. Outro exemplo, desta vez relativo à economia de energia, é o de um sistema que monitora a presença das pessoas na residência e desliga equipamentos quando estes não estão sendo utilizados (no caso mais simples, lâmpadas).

Note-se que, neste último exemplo, os mesmos sensores que são utilizados para detectar presença nos ambientes podem ser utilizados para o sub-sistema de segurança residencial. Efetivamente, um sistema de automação bem concebido evita desperdícios e conseqüentemente gera economias. Neste mesmo sentido, o fato de poder comandar sistemas de *Home Theater*, *DVD*, TV e ar-condicionado através de um aplicativo em um *smartphone* também proporciona conforto e economia, evitando a utilização de quatro controles-remotos independentes.

O mercado de automação residencial é um mercado em consolidação. As empresas e tecnologias existentes ainda não estão estabelecidas. Algumas empresas, porém, se destacam com soluções robustas e, em muitos casos, interoperáveis. A seguir são apresentadas algumas das soluções com maior aceitação pelos usuários.

## 2.2 SCHNEIDER ELECTRIC

A Schneider Electric SA é uma empresa francesa, sediada em Rueil-Malmaison, especializada em produtos para sistemas de distribuição, automação e gerenciamento de energia. No ramo de automação residencial, ela oferece duas soluções diferentes:

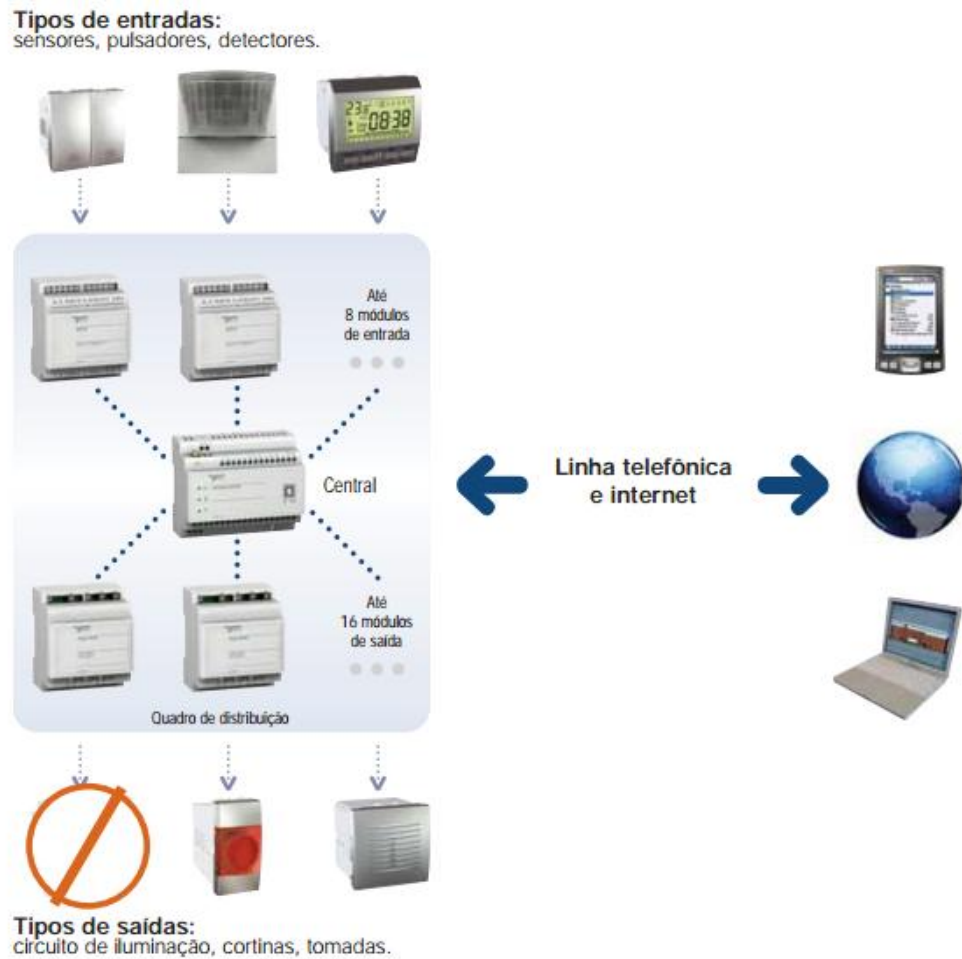
- Linha IHC ou Intelligent Home Control
- Linha Bus KNX

Apesar de resultarem em benefícios similares para os usuários, tecnicamente elas são constituídas de maneira diferente. Além disso, devido à maneira como foram projetadas, a solução IHC é mais adequada para pequenas redes de automação, ao passo que a solução KNX é adequada para automações residenciais ou prediais de grande porte.

### **2.2.1 Linha IHC - Intelligent Home Control**

A linha IHC é uma linha simples tecnicamente. Seus dispositivos são divididos em entradas e saídas. Cada entrada é conectada a um módulo de entrada, que pode receber até 16 entradas diferentes. Cada saída é conectada a um módulo de saída, que pode controlar até 8 saídas digitais ou dimmerizadas. Os módulos de entrada/saída são multiplexadores/de-multiplexadores dos sinais que são levados à central. A central coordena a lógica de acionamento das saídas a partir das entradas, conforme figura 1. Os módulos são alimentados individualmente.

Figura 1 – Linha IHC da Schneider Electric



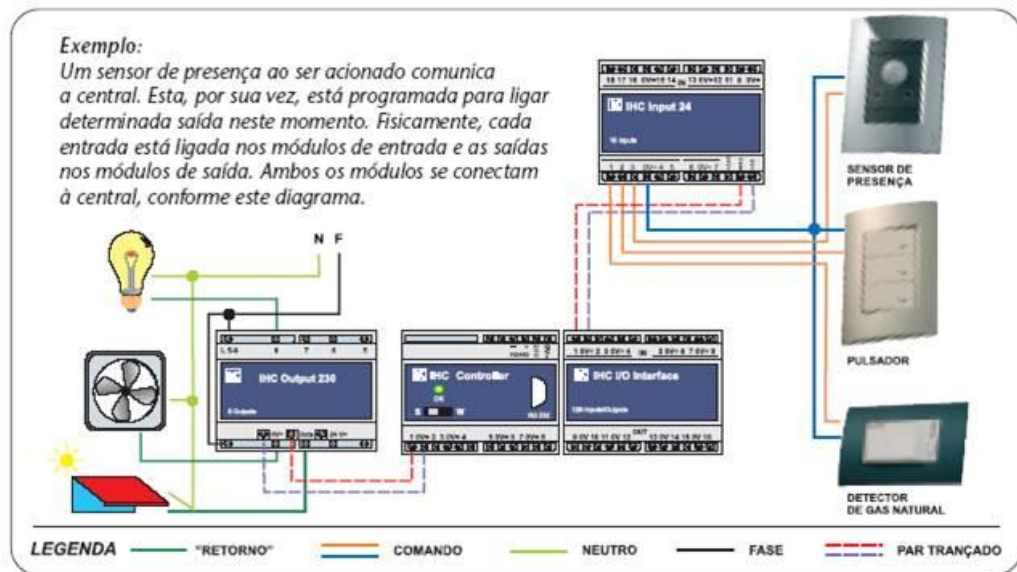
Fonte: Manual Prime IHC, pg. 9 – Schneider Electric

A central pode ser conectada à internet ou linha telefônica, o que permite o acionamento e monitoramento dos dispositivos remotamente, através de laptops e PDAs. Os módulos de entrada/saída e a central são instalados no quadro de distribuição da casa. A central é programada através de um software fornecido pela fabricante e pode ser facilmente reconfigurado para gerar novas lógicas de controle.

A linha apresenta uma ampla gama de sensores e atuadores, podendo ser integrada com leitores biométricos, dimmerizadores, controles remotos centralizadores, sensores de luminosidade e presença, entre outros da linha Schneider Electric. Por outro lado, a grande desvantagem deste sistema é a necessidade de passar uma grande quantidade de cabos, o que impacta no custo de instalação e manutenção do sistema, como pode ser observado na figura 2.



Figura 2 – Instalação linha IHC da Schneider Electric



Fonte: [http://www.aspro.eng.br/automacao\\_conceito.php](http://www.aspro.eng.br/automacao_conceito.php)

Como fica claro no exemplo da figura 2, cada sensor/atuador deve ser ligado com um cabo independente aos módulos de entrada/saída, necessitando uma grande quantidade de cabos. O sistema BUS, descrito a seguir, contorna este problema de maneira elegante, além de trazer outros benefícios. Pelos motivos acima expostos, esta configuração só se mostra adequada/viável, para pequenos sistemas de automação.

### 2.2.2 Linha BUS KNX

O protocolo KNX é único padrão aberto (EN 50090, ISO/IEC 14543-3) para automação residencial/predial. Ele se baseia no modelo conceitual OSI de redes na sua definição, de forma que pode ser implementado de maneira independente do suporte físico (par trançado, ondas de rádio, etc).

O protocolo herdou conceitos de três outros padrões em automação residencial bem difundidos (em especial na Europa) : European Home Systems Protocol (EHS), BâtiBUS e European Installation Bus (EIB). A KNX Association tem a missão de desenvolver e promover o KNX como um standard mundial e possui como membros diversas empresas de grande porte como Schneider Electric, ABB e SIEMENS.

A grande vantagem do sistema KNX da Schneider Electric sobre o sistema IHC é que o primeiro se comunica em uma topologia de rede tipo BUS. Esta

topologia propicia ganhos tanto pela facilidade de instalação quanto de reconfiguração e manutenção.

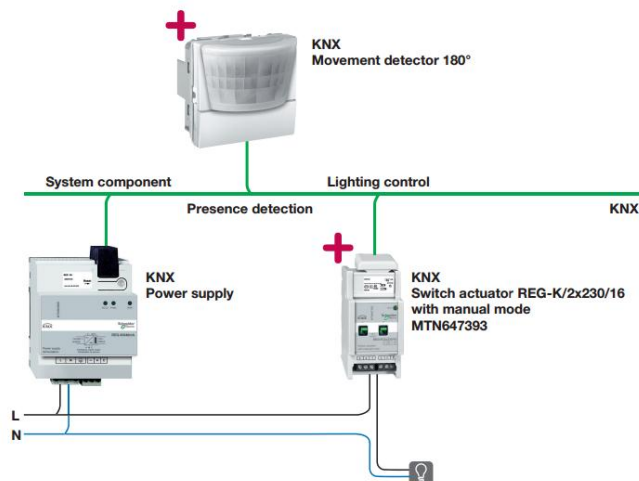
Além disto, pelo fato do protocolo KNX ser um padrão aberto, contando com diversas empresas associadas no mercado, os equipamentos desse sistema são interoperáveis em relação a equipamentos de outras fabricantes. Este ponto é extremamente importante na perspectiva do usuário, por garantir que seu sistema não ficará obsoleto no curto/médio prazo e ao mesmo abrir uma ampla gama de escolha de dispositivos/fabricantes para expansão da sua rede de automação.

Mais especificamente, a linha BUS KNX da Schneider Electric oferece os seguintes produtos (Schneider Electric, 2011):

- Componentes de sistema (fontes de alimentação, acopladores, etc)
- Interfaces / Gateways
- Pulsadores
- Módulos de entrada binários
- Sensores de presença e movimento
- Estações meteorológicas
- Módulos comutadores
- Módulos para cortinas e persianas
- Módulos de dimerização
- Controles remotos e displays
- Controle de temperatura

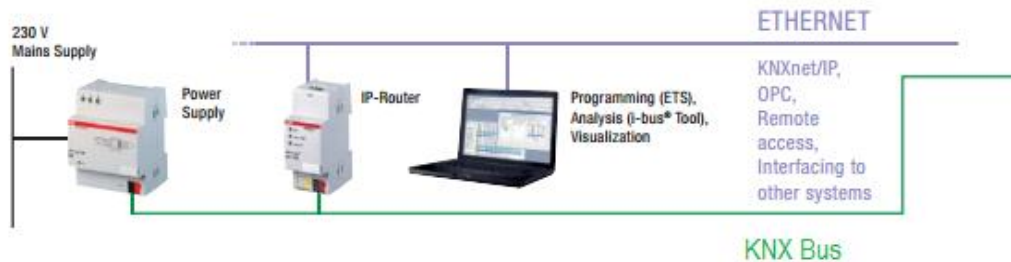
Um exemplo de uma instalação para controle de acionamento de uma lâmpada através de detecção de movimento é mostrado na figura 3.

Figura 3 – Linha BUS KNX –Schneider Electric



O exemplo acima mostra a simplicidade de instalação deste sistema, dado que todos os dispositivos utilizarem uma mesma via de comunicação. Os dispositivos são configurados por um software fornecido pela própria KNX Association, denominado ETS. Esta mesma ferramenta é compatível com todos os outros fabricantes que desenvolvem produtos baseados nas especificações KNX.

Figura 4 – Configuração BUS KNX



Fonte: ABB i-bus® KNX Smart Home and Intelligent Building Control Product Range Overview 2013

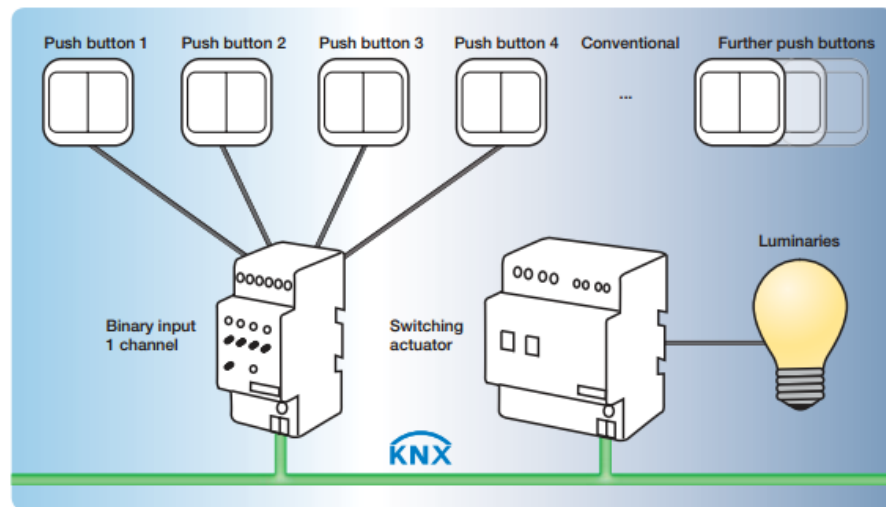
Além desta ferramenta oficial, diversos outros softwares proprietários e aplicativos existem no mercado para interação, análise e configuração de redes KNX. Como pode ser visto na figura 4, um computador normal pode se comunicar com a rede KNX através de um gateway Ethernet/KNX, neste exemplo denominado IP-Router.

### 2.3 ABB

A ABB é uma fabricante europeia de soluções para sistemas de potência, automação e robótica e figura entre as maiores empresas do mundo em termos de faturamento. O sistema de automação residencial/predial oferecido por esta fabricante é denominado ABB i-bus, e também utiliza o protocolo padronizado KNX (ABB. 2010).

A figura 5 apresenta um diagrama simplificado de ligação de um sistema de acionamento de lâmpadas. Assim como o sistema da Schneider Electric, os dados trafegam por um BUS comum de comunicação, utilizando o protocolo KNX. Desta forma, ambos os sistemas podem funcionar de maneira interligada ou podem ser combinados para formar um único sistema composto por dispositivos mistos.

Figura 5 – Linha i-bus - ABB



Fonte: ABB i-bus® KNX Application Manual Lighting, pg. 16. - ABB

## 2.4 SIEMENS

Um solução oferecida pela Siemens é o microPLC Logo!. Essencialmente ele é um pequeno Controlador Lógico Programável (CLP) que pode ser utilizado para tarefas de automação residencial. Por não ser um dispositivo feito sob medida para aplicações em automação residencial, ele pode demandar mais tempo de trabalho para configuração da rede. Por outro lado, para automações pontuais, se apresenta como uma solução interessante financeiramente. Este microPLC pode ser visto na figura 6.

Figura 6 – LOGO! microPLC

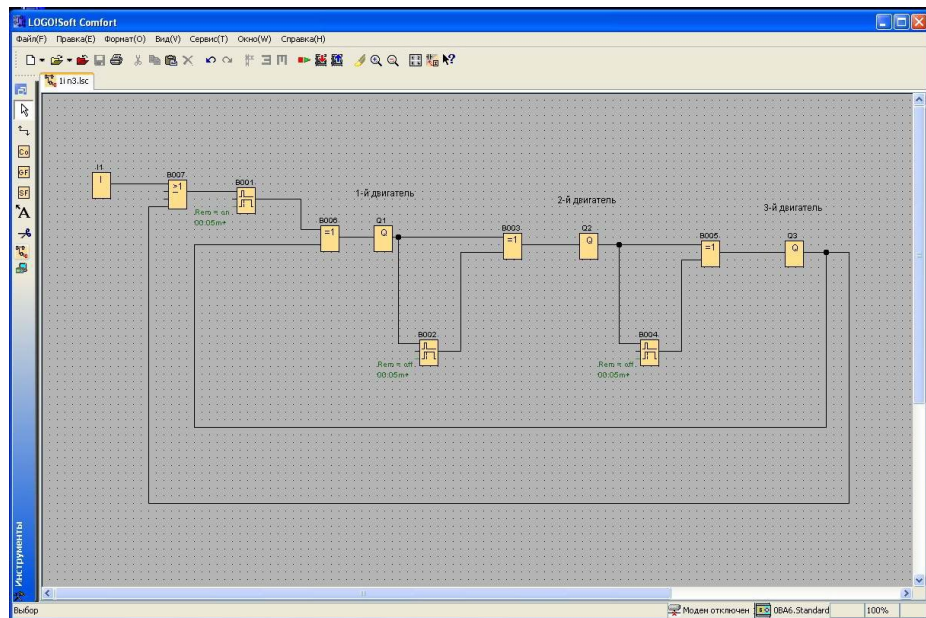


Fonte: LOGO! App V1.0.0 – Siemens

Através de conexões Ethernet, este microPLC pode se comunicar com outros microPLCs da mesma família, formando uma rede de automação ou pode ser programado via *PC*. Também é possível programá-lo via interface USB. O software é fornecido pela Siemens e se chama Logo!Soft Comfort.

O microPLC Logo! pode ser programado com duas linguagens de programação gráficas essencialmente equivalentes: Ladder ou Diagrama de Blocos Funcionais (FBD) (PRUDENTE, Francesco). A figura 7 apresenta um exemplo de programação do microPLC via FBD no Logo!Soft Comfort.

Figura 7 – Logo!Soft Comfort



Fonte: LOGO! App V1.0.0 - Siemens

## 2.5 OUTROS SISTEMAS E PROTOCOLOS

As soluções apresentadas até o momento são amplamente empregadas no mercado Europeu. Nos Estados Unidos, alguns outros protocolos tiveram uma boa aceitação pelo mercado.

Entre eles, encontra-se o protocolo X10, baseado em tecnologia PLC (*Power Line Communication*). Este protocolo foi originalmente desenvolvido pela Pico Electronics, em 1975. Neste padrão, os sinais de comando do sistema de automação são transmitidos diretamente sobre os condutores de energia do domicílio.

Mais especificamente, quando a senóide da tensão de alimentação cruza por

0V, um bit de informação é transmitido em uma frequência de rádio – geralmente 120kHz. Esta frequência pode ser posteriormente filtrada pelo aparelho receptor, eliminando facilmente a componente em 60Hz da rede elétrica. Por transmitir apenas um bit por cruzamento de zero da tensão e necessitar ainda de enviar bits redundantes para correção de erros, este protocolo apresenta uma taxa de comunicação muito baixa, ficando na ordem de 20bits/s (X10, 2010).

Para algumas aplicações, esta taxa é aceitável. Para modernos sistemas de automação predial/residencial, porém, esta característica é muito limitante. Além disto, a presença de ruídos na rede ou mesmo falta de energia podem paralisar o funcionamento do sistema. Em outras palavras, ele não é adequado para sistemas críticos, como o caso de sistemas de segurança.

Outro sistema alternativo para automação residencial é o LonWorks, da Echelon Corporation. A tecnologia da LonWorks, baseada no protocolo desenvolvido referido como LonTalk, foi concebida mais especificamente para atender necessidades de sistemas de controle. A tecnologia está pautada em 18 requisitos de sistema, que satisfazem as necessidades de sistemas de controle em relação a redes. Um exemplo de requisito, de acordo com (LONWORKS – ECHELON, 2013), é o que segue:

*It must be possible to engineer the control network so that the real-time requirements of the application are met. This involves:*

- *Designing the network to meet response time criteria by limiting the number of nodes per link, and tuning the communications so the network will not become overloaded.*
- *Specifying that a given communications transaction will either succeed or fail within a specified time, with the success or failure of that transaction known to the application.*

***Rationale:*** *In a control system, a late packet would result in some node not doing its function in synchronization with the other nodes.*

Em outras palavras, para aplicações de tempo-real e controle, o sistema deve se comportar da maneira mais determinística possível. Um *jitter* elevado em uma malha de controle pode ocasionar comportamentos indesejável, quicça catastróficos a nível de performance do sistema. Ao menos as características de pior caso devem ser conhecidas, o que não é garantido em muitos protocolos. As aplicações do LonWorks vão de *Smart Buildings a Smart Cities*, passando por *Smart Grids e Smart Control*.

Além destas tecnologias, uma vasta gama de protocolos e sistemas estão

disponíveis no mercado, como é o caso da fabricante americana *Control4* ou a brasileira Neocontrol. Ambas apresentam gamas de dispositivos interessantes, com funcionalidades que por vezes outros sistemas não atendem.

Este é o caso do Cubo, feito em aço inox e acrílico injetado, da Neocontrol que pode ser visto na figura 8. Ao girar o Cubo, alterando a face que fica voltada para cima, é possível alterar a configuração do jogo de luzes de uma sala, por exemplo. Outra possibilidade seria alterar o estilo musical através do *setlist* que esta sendo reproduzido pelo sistema de som da residência.

Figura 8 – Cubo de controle Neocontrol



Fonte: <http://www.neocontrol.com.br/public/cubo>

Uma característica comum, que permeia quase todos os sistemas presentes no mercado, é a modularidade. Por serem concebidos desta forma, a sua expansão ou manutenção se torna fácil. De fato, tecnicamente a identificação de falhas e posterior reparo são feitos de maneira ágil. Através da troca do módulo, o sistema do cliente volta a se tornar operacional, ao passo que o reparo/análise em nível de circuito da peça defeituosa pode ser feito no laboratório da empresa desenvolvedora.

A grande desvantagem deste mercado *hiper* segmentado é a pluralidade de protocolos e soluções existentes. Por consequência, a interoperabilidade não é garantida na grande maioria dos casos. Pontualmente, existem *gateways* que possibilitam a integração de tecnologias diferentes, em detrimento da performance dos sistemas.

Neste contexto de variedade de soluções, a padronização de protocolos se mostra essencial para um crescimento ordenado, permitindo uma posterior integração *Smart Buildings-Grids-Cities*. O KNX se mostra como uma primeira tentativa de padronização, apesar de haver um longo caminho ainda para ser considerado padrão de mercado.



### 3 SISTEMA DESENVOLVIDO: XHOME

Após análise das soluções existentes no mercado para automação residencial, um sistema próprio foi concebido. Como será visto no decorrer deste capítulo, esta solução apresenta características competitivas em relação aos sistemas de mercado, por ser acessível financeiramente e de fácil instalação.

#### 3.1 ARQUITETURA E FUNCIONALIDADES

O sistema desenvolvido, daqui para frente referido como Xhome, foi projetado para atender algumas funcionalidades básicas. A definição destes requisitos assegura que a solução proposta atenderá necessidades reais de usuários de sistemas de automação residencial.

Primeiramente, deseja-se que ele tenha uma interface com a internet, permitindo a atuação e o monitoramento remoto da residência. Em segundo lugar, o sistema deve ser “escalável”, ou seja, deve permitir um crescimento uniforme, a partir da replicação da ideia base por trás da proposta. Além disto, é desejável que o sistema seja de fácil instalação, baixo consumo de energia e, preferencialmente, de baixo custo.

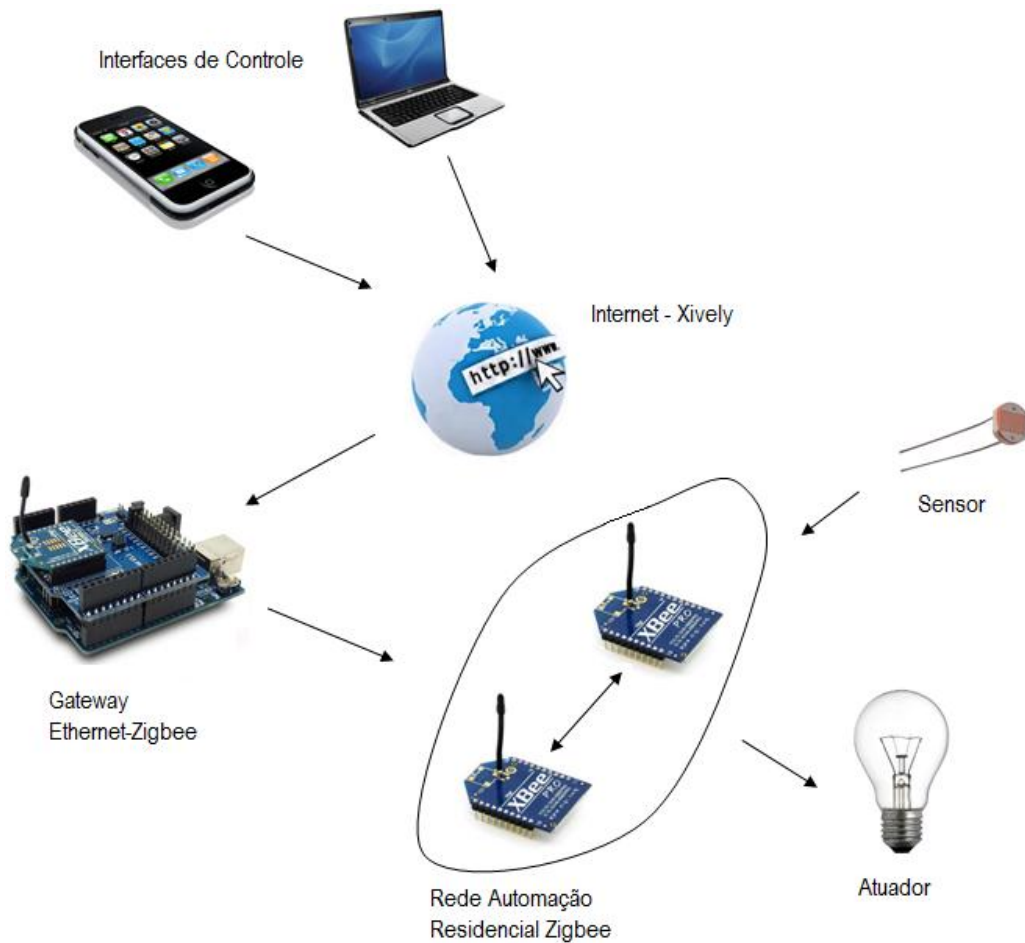
Com vistas a atender esses requisitos, diversas soluções foram estudadas. A arquitetura utilizada foi a que apresentou melhor aderência a esses requisitos básicos, além de possuir uma série de outros benefícios, como serão mostradas no decorrer deste capítulo. Ela é mostrada na figura 9.

De uma maneira geral, o sistema desenvolvido pode ser dividido em 4 partes:

- Interface de controle;
- Um servidor internet que gerencia o armazenamento das informações do sistema;
- Um gateway Ethernet/Zigbee;
- Sensores e atuadores espalhados pela casa.



Figura 9 – Arquitetura sistema Xhome



Fonte: Elaborada pelo autor

A interface de controle utilizada é um *tablet*, *laptop* ou *smartphone*. Mais genericamente, qualquer dispositivo com interface com a internet pode ser utilizado para acessar o sistema.

Para o servidor de armazenamento de dados, diversos sites alinhados ao conceito de “Internet of Things” disponibilizam soluções robustas e gratuitas. O sistema escolhido para utilização no Xhome é denominado Xively (XIVELY 2013), antigamente conhecido como Cosm. Este site é um dos mais populares neste mercado. Ele apresenta um série de facilidade de utilização e possibilidade de conexão com outros sites, motivos pelo qual foi escolhido para a implementação do Xhome. Mais detalhes sobre o site serão apresentados no decorrer deste capítulo.

O gateway Ethernet/Zigbee, por sua vez, possui duas funções principais: coordenar a rede, centralizando as informações dos sensores/atuadores, e fazer a interface entre o protocolo zigbee (rede de automação doméstica) e a internet (meio físico ethernet). Zigbee é o nome de um protocolo de comunicação *wireless* que

apresenta características muito apropriadas para redes de automação residencial, como será descrito em mais detalhes em um capítulo posterior.

O Xbee (solução técnica que implementa o protocolo Zigbee) e o microcontrolador utilizado neste trabalho utilizam pouca potência. Além disto, o Xbee possui um modo denominado *sleep* que reduz o consumo de corrente para alguns micro-ampéres, permitindo que ele opere ligado em uma bateria por mais de um ano, sem recarregá-la. Estas escolhas de hardware permitem que o sistema seja instalado sem passar fiação (por ser um sistema *wireless*) e isenta a necessidade de conectar sensores/atuadores na linha de alimentação elétrica da residência, o que pode ser muito vantajoso em algumas situações. Alguns exemplos são sensores/atuadores instalados em jardins desprovidos de fiação elétrica ou sensores/atuadores móveis.

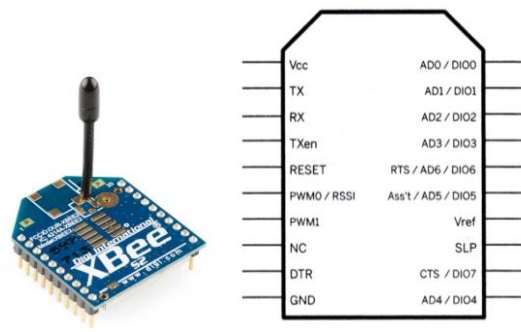
Por último, a rede de automação interna da residência, como já foi mencionado, é baseada no protocolo de rede Zigbee, implementado nos dispositivos denominados Xbee.

### 3.2 REDES XBEE: CONCEITO E APLICAÇÃO

O padrão Zigbee é uma especificação para dispositivos de comunicação wireless que demandam baixas velocidades de transmissão, baixo consumo de energia e segurança. Está baseado no *standard* 802.15.4 do IEEE, o mesmo que dá origem a outros LR-WPAN (*low-rate wireless personal área networks*) como o WirelessHART (ZIGBEE, 2013). As características desta especificação se adaptam muito bem a aplicações em automação residencial, por não demandarem altos fluxos de informações, porém necessitando conexões seguras e, preferencialmente, dispositivos que consumam pouca potência.

O dispositivo selecionado para implementar o protocolo Zigbee é o hardware denominado Xbee, versão S2, da fabricante Digi International®. A figura 10 apresenta o hardware e suas conexões.

Figura 10 – Modulo Xbee e seus pinos correspondentes



Fonte: Elaborada pelo autor

O Xbee é alimentado em 3.3V e possui diversas entradas e saídas digitais de uso geral. As saídas digitais podem ser utilizadas para acionar dispositivos eletrodomésticos em residências, como no caso deste trabalho. Além disto, possui entradas ligadas a conversador ADC de 10 bits para amostragem de sinais analógicos. Por último, possui um canal de comunicação USART (TX/RX) cuja informação é retransmitida wireless via protocolo Zigbee.

A frequência de operação do dispositivo é de 2.4 GHZ e trabalha com até 250kbps de taxa de transferência. Pode alcançar distâncias de comunicação entre 2 dispositivos em campo aberto de até 100m. Uma versão mais potente, denominada de Xbee PRO, chega a alcançar mais de 1Km em ambiente externo. Uma das vantagens das redes Xbee é que cada dispositivos da rede serve como um retransmissor do sinal. Desta maneira, o raio de comunicação das redes pode ser facilmente expandido, cobrindo todos os ambientes da residência onde o sistema esta sendo implantado, no caso de automação residencial.

### 3.2.1 Características técnicas de redes Xbee

Cada dispositivo Xbee pode ser configurado com um dos 3 perfis padrões que formam as redes Xbee: coordenador (*Coordinator*), roteador (*Router*) ou dispositivo final (*End Device*). Cada perfil de configuração atribui um certo número de características aos dispositivos, que irão exercer papéis diferentes dentro da rede.

O dispositivo Coordenador é responsável pela inicialização da rede, distribuição de endereços, manutenção, reconhecimento de todos os nós, entre outras funções, podendo servir como ponte entre várias outras redes ZigBee (DIGI INTERNATIONAL INC, 2010). No caso do sistema desenvolvido, este dispositivo se

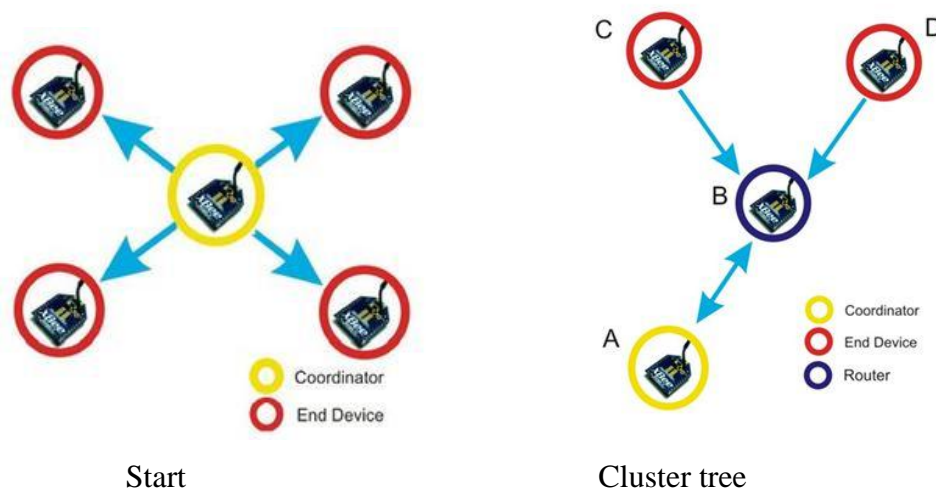
encontra no gateway Zigbee/Ethernet. Toda rede Xbee possui um e apenas um dispositivo Coordenador. Por possuir tarefas essenciais de manutenção da rede, este dispositivo não pode entrar em modo *Sleep*, como acontece na configuração *End Device*.

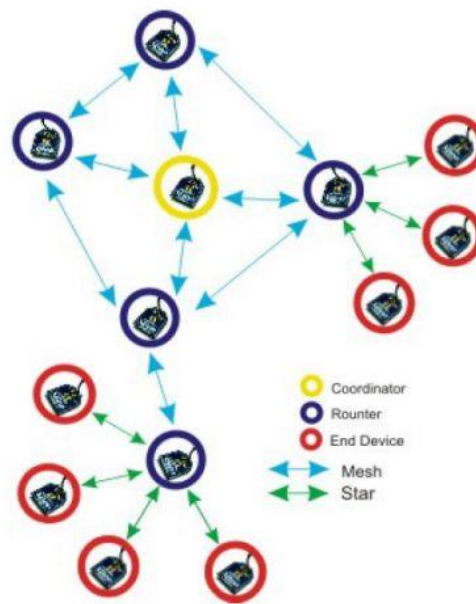
Na configuração roteador, o dispositivo tem as características de um nó normal na rede, mas com poderes extras para exercer a função de roteador intermediário entre nós, sem precisar do Coordenador. Por intermédio de um roteador uma rede ZigBee poder ser expandida, e assim ter mais alcance. No caso de automação residencial, um roteador pode ser usado para amplificar o sinal da rede entre andares de um prédio.

Por último, na configuração dispositivo final de rede (End Device) o dispositivo recebe e envia informações à rede, sem precisar estar acordado todo o tempo. No caso eventual de um roteador tentar enviar informações para um End Device que não se encontra acordado, este irá armazenar as informações em uma memorial interna e tentará a conexão posteriormente. Os dispositivos de final de rede, quando operando em modo *sleep*, consomem apenas alguns micro-ampéres, podendo operar por mais de um ano conectados a baterias/pilhas. Por terem estas características, são os dispositivos utilizados para serem ligados em sensor/atuadores que se encontram nas pontas da rede.

Com base nos 3 tipos de dispositivos existentes descritos acima, podemos obter diferentes topologias de rede, conforme a figura 11.

Figura 11 – Topologias rede Xbee





Star/Mesh

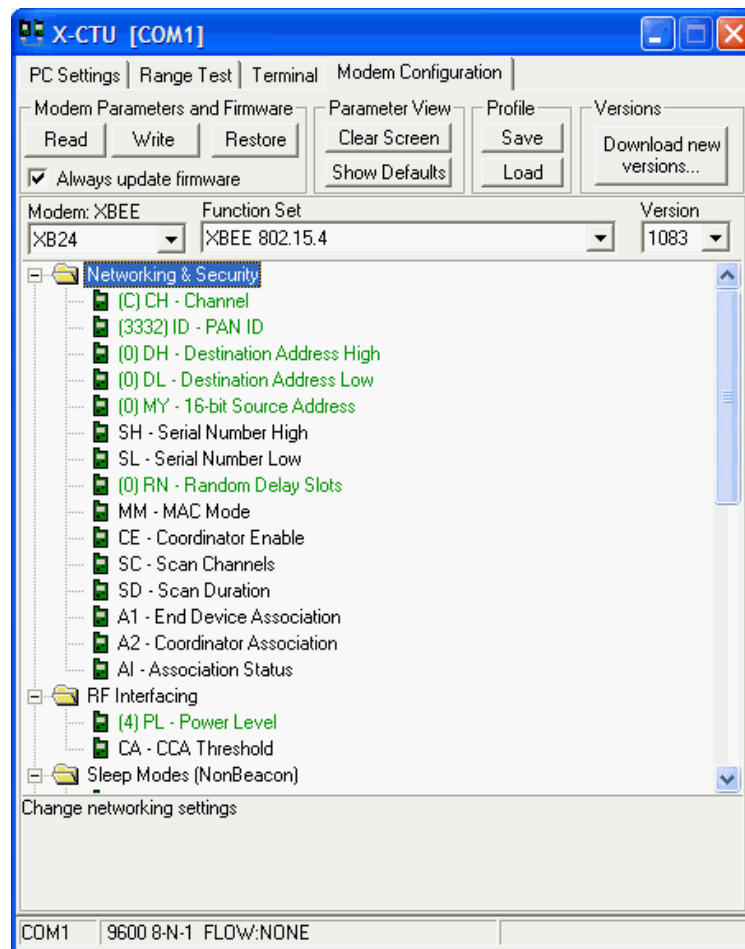
Fonte: <http://xbee.wikispaces.com/MESH>

A configuração que será obtida pela rede depende da maneira como os dispositivos foram configurados (coordenadores, roteadores, etc) e da localização geométrica dos mesmos. Cabe aos dispositivos, conforme especificado no padrão Zigbee, determinar a melhor topologia a ser utilizada, de maneira a otimizar a distribuição dos pacotes na rede.

A configuração do Xbee quanto ao papel que ele exerce na rede, entre outros aspectos, é feita através da sua porta USART. A maneira padrão para acessá-la através da utilização de um computador e um conversor USB/USART. Um software gratuito desenvolvido pelo fabricante é disponibilizado para facilitar a tarefa. Este software se chama X-CTU e é mostrado na figura 12.

Através do X-CTU, podemos atualizar o firmware do dispositivo, alterar o papel que exerce na rede, alterar seus parâmetros (como criptografia de rede e nome da rede), fazer testes de rede, entre outras possibilidades.

Figura 12 – X-CTU

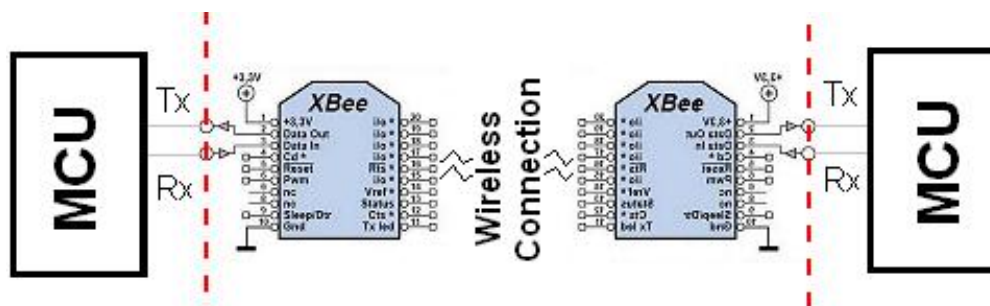


Fonte: Elaborada pelo autor

### 3.2.2 Modos de operação do Xbee

A configuração de conexão mais utilizada para os módulos Xbee é como apresentada na figura 13, onde MCU representa microcontrolador/microprocessador.

Figura 13 – Comunicação Xbee/MCU



Fonte: Elaborada pelo autor

Os módulos Xbee S2 possuem 2 modos de operação:

- Modo transparente
- Modo API (Application Programming Interface)

No seu modo de operação mais simples, o modo transparente, toda informação recebida na porta USART é transmitida via Zigbee aos outros módulos Xbee conectados na rede. Na prática, tudo se passa como se os módulos estivessem conectados entre si por conexões USART.

Apesar deste modo ser de fácil utilização e ser suficiente para um determinado número de aplicações, ele apresenta fortes restrições. Um dos problemas encontrados é que, antes da formação da rede, cada dispositivo deve ser pré-configurado com a informação de para qual dispositivo ele enviará as mensagens. O envio de mensagens diferentes para 2 ou mais dispositivos diferentes não é possível, pois o destinatário das mensagens é fixo. Além disto, operações como requisição de status, leitura de valores analógicos de outros dispositivos, entre outras, não são possíveis. Por estes e outros motivos, o coordenador da rede de automação desenvolvida não opera neste modo, tendo em vista que ele precisa enviar pacotes para diferentes dispositivos, requisitar status, etc.

O modo API é mais completo, porém mais complexo de se utilizar. Neste modo, as informações recebidas na porta USART do dispositivo não são automaticamente transmitidas via *wireless*. Do contrário, as informações recebidas são interpretadas como *frames* (FALUDI, 2010). Cada *frame* é identificado por um número (*API identifier*) e executa uma determinada ação. Os tipos principais são listados na tabela 1.

Tabela 1 – Principais tipos de *frames* Xbee

<b>Identificador do Frame</b>	<b>Descrição do Frame</b>
0x08	Comando AT imediato
0x09	Comando AT (fila)
0x17	Requisição de comando remoto
0x88	Resposta de comando AT
0x8A	Status do Modem
0x10	Requisição de TX
0x8B	Resposta de TX

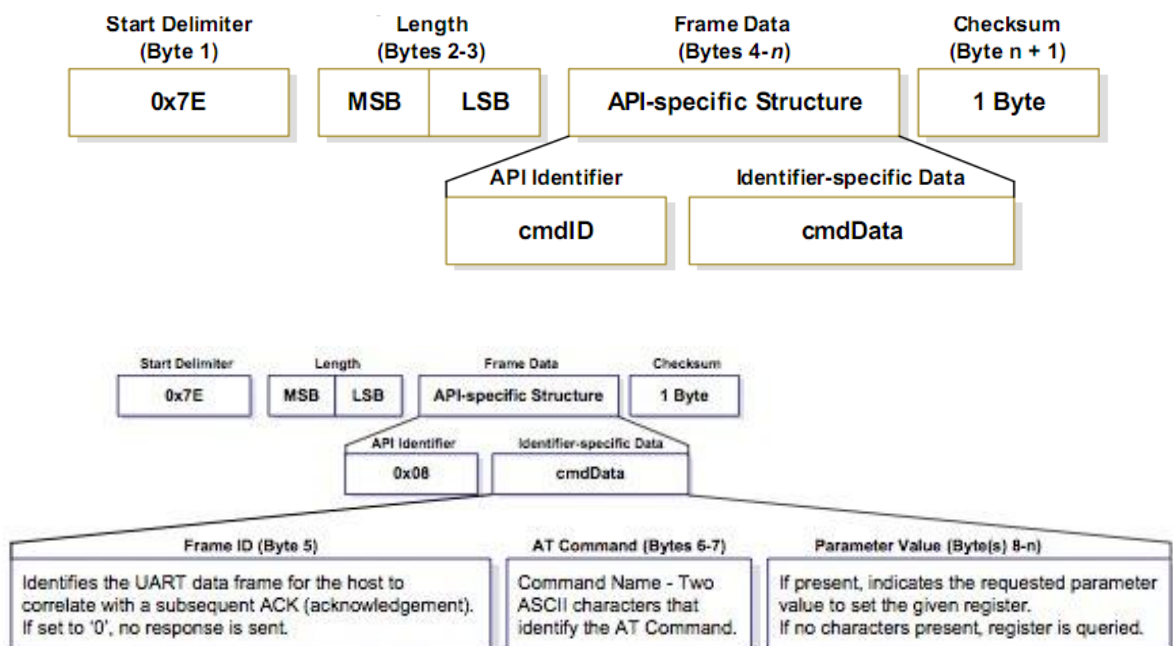
0x90	Recebido RX
0x92	Recebida informação de I/O
0x95	Identificador de nó
0x97	Resposta de comando remoto

Fonte: Elaborada pelo autor

Sem entrar em detalhes, estes frames abrem uma série de possibilidades que não estão disponíveis no modo transparente. Entre elas, é possível requisitar status de dispositivos, reconfigurar os dispositivos *over-the-air*, requisitar informações e comandos remotos, etc. Com estes *frames*, tem-se plena autonomia para gerenciar a rede de automação residencial.

A estrutura básica e o desdobramento para um tipo de *frame* específico é mostrado na figura 14.

Figura 14 – Estrutura de frame modulo Xbee



Fonte: <http://rubenlaguna.com/wp/2009/03/12/example-of-xbee-api-frames/>

Os comandos AT são uma espécie de *instruction set* para comunicação e configuração dos módulos Xbee. Eles podem ser divididos nas seguintes classes:

- Comandos especiais
- Comandos de rede e segurança
- Interface RF



- Sleep (Low Power)
- Interface Serial
- Configurações de I/O
- Diagnóstico de rede
- Comandos de opções AT

Por último, vale ressaltar que dois dispositivos de uma mesma rede não precisam necessariamente funcionar no mesmo modo de operações. Com efeito, no caso do Xhome, o coordenador da rede funciona em modo API (pois precisa requisitar status e se comunicar com diversos dispositivos) e os End Devices estão em modo transparente, pois não necessitam de funcionalidades avançadas e, neste modo, são mais facilmente configuráveis.

### 3.2.3 Configurações dos dispositivos na rede Xhome

No sistema desenvolvido, estão sendo utilizados 3 Xbees:

- 1 Xbee ligada ao Gateway Ethernet/Zigbee.
- 2 Xbee ligados nos atuadores/sensor, sendo um por placa.

Todos os dispositivos foram configurados para operar na rede de ID 3332 e canal 0, de forma a garantir que todos estejam operando na mesma rede/canal. Os valores utilizados são valores padrões de referência em manuais da fabricante Digi.

O Xbee ligado ao *gateway* Ethernet/Zigbee foi configurado em modo coordenador API. Sendo o coordenador da rede, ele é responsável por criar a rede inicialmente, distribuir IDs para os nós, entre outros. Efetivamente, o *gateway* está sendo utilizado como dispositivo central da rede de automação, motivo pelo qual foi configurado desta forma.

Por outro lado, o modo API foi selecionado por ser o único modo que atende às necessidades do sistema. O modo transparente não permite o envio de pacote de configuração de saídas digitais remotas, como os pacotes que serão utilizados para acionar os atuadores no sistema. Além disto, ele também não recebe informações de amostras digitais/analógicas dos outros nós da rede. Na realidade, o modo transparente somente permite que o fluxo de informação USART de um Xbee seja visto por outro Xbee, sendo os endereços de origem e destino fixados *a priori*. O fato

de impossibilitar a troca do endereço de destino dos pacotes também é limitante, por não permitir a comunicação com dois dispositivos diferentes de maneira independente.

Os dispositivos Xbees ligados nas placas de atuação/sensoriamento foram configurados no modo roteadores AT. Estes foram configurados em modo roteadores para permitirem o roteamento de pacote de um para o outro, dependendo da maneira como forem instalados fisicamente. Na realidade, se for conhecida a distribuição destes dispositivos fisicamente, podemos utilizar igualmente o modo End Device AT, sem alterações no resto do sistema. Neste caso, o dispositivo ainda apresentaria a vantagem de poder ser configurado em modo *sleep* e consequentemente consumir pouca energia. Um dispositivo neste modo pode ser usado como um sensor remoto dentro da residência, como um sensor de luz, que não necessita ser ligada na rede elétrica e cuja bateria pode vir a durar mais de um ano sem interrupção.

O pino D1 dos Xbees dos atuadores/sensores como entradas analógicas (ADC 10 bits). Estas entradas são amostradas a cada 1.5s, enviando o valor com esta mesma periodicidade para o coordenador da rede, no caso o *gateway* Ethernet/Zigbee. Além disto, os pinos D0 foram configurados como saídas digitais, permitindo o controle de dispositivos externos, os atuadores.

Nestes Xbees, a escolha pelo modo AT foi por simplicidade de configuração e por não necessitarem a utilização de comandos mais avançados do modo API. De fato, estes dispositivos sempre enviarão os seus pacotes para o mesmo destino: o Coordenador da rede. Além disto, eles não controlam nenhum nó da rede, apenas enviam informações e são controlados. Por estes motivos, a utilização do modo API não se justificava.

No caso do sistema aqui apresentado, estão sendo utilizadas apenas duas placas de monitoramento/atuação. O objetivo sendo demonstrar que o sistema permite comandar e receber informações de placas distintas de maneira independente. Por outro lado, o número de nós da rede pode ser aumentado facilmente, apenas replicando a configuração dos Xbees das placas de atuação/monitoramento. A ideia por traz do sistema é que cada cômodo da residência possa ser automatizado por uma célula de monitoramento/atuação – a célula desenvolvida.

### 3.3 GATEWAY ETHERNET/ZIGBEE

O *gateway* Ethernet/Zigbee desenvolvido é constituído de 3 partes básicas:

- Arduino Uno (ARDUINO UNO, 2013)
- Shield Ethernet para Arduino
- Placa de interface Arduino/Xbee

A arquitetura de comunicação entre estes componentes é mostrada na figura 15.

Figura 15 – Protocolos utilizados no Gateway



Fonte: Elaborada pelo autor

Como podemos ver, o Arduino consegue enviar e receber dados da internet através da interface SPI do Shield Ethernet e é capaz de enviar e receber dados da rede de automação residencial através da interface USART do Xbee coordenador da rede.

A seguir será apresentada uma breve descrição destes três hardwares utilizados no trabalho.

#### 3.3.1 Arduino Uno

O Arduino é um Kit de desenvolvimento microcontrolado *open-source* desenvolvido para aplicações com baixa demanda de processamento. Atualmente está disponível em várias versões (Uno, Nano, Mega, etc) onde basicamente as alterações são a nível de microcontrolador utilizado e de tamanho da PCB.

Para integrar o sistema Xhome, foi escolhido o modelo Arduino Uno. Esta versão utilizada o processador de 8 bits ATmega328 da fabricante Atmel. As principais características técnicas deste microcontrolador são listadas abaixo:

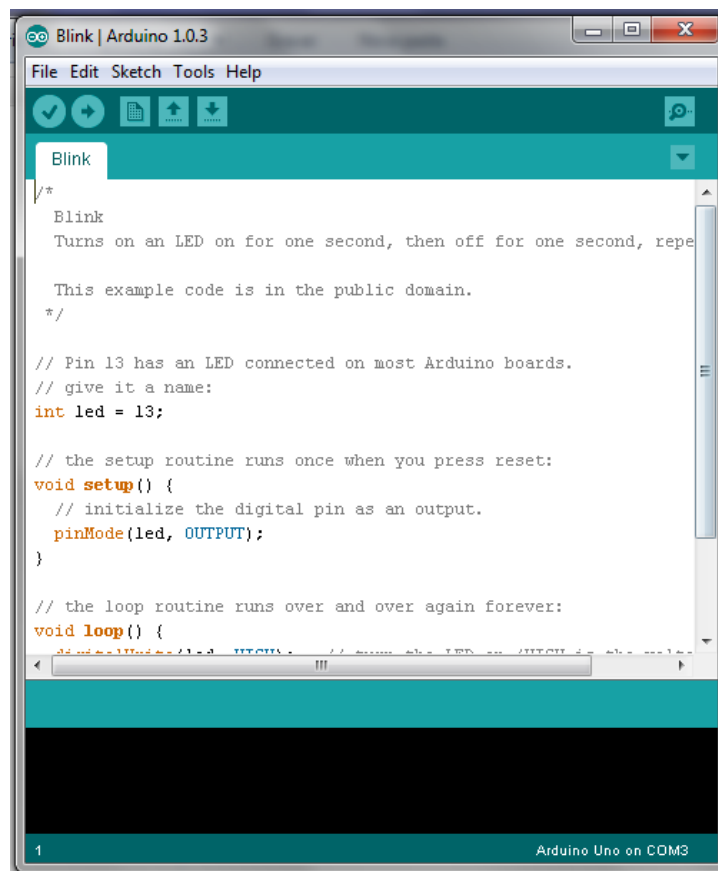
- AVR – Arquitetura RISC tipo Harvard modificada
- 20 MHz

- 32K Bytes de memória de programa flash
- 6 ADCs de 10 bits
- Interfaces USART e SPI
- 6 modos de Sleep diferentes
- Diversas GPIOs

O hardware utilizado possui todas as interfaces de comunicação necessárias, além de possuir modos de *sleep*, adequados para poupar energia em sistemas econômicos.

O kit Arduino pode ser programado através de um cabo USB conectado a um computador. O software utilizado para programação é fornecido gratuitamente pelo fabricante e pode ser executado em ambiente Windows, Mac OS X ou Linux. Possui uma linguagem de programação específica para a plataforma, porém muito similar a C. O software de programação do Arduino é mostrado na figura 16. Este mesmo software pode ser utilizado para os diversos tipos de Arduino existentes no mercado, conforme mencionado acima.

Figura 16 – Arduino IDE 1.0.3

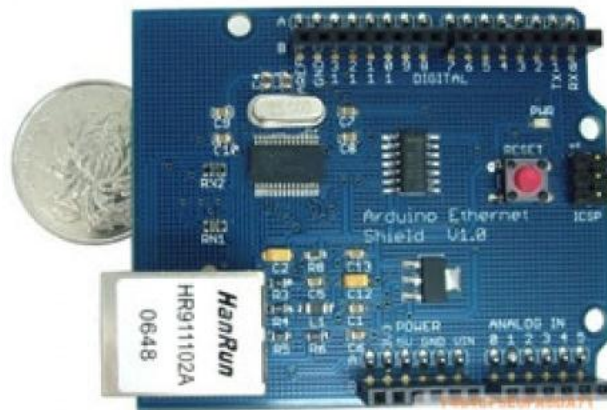


Fonte: Elaborada pelo autor

### 3.3.2 Shield Ethernet para Arduino

O Shield Ethernet utilizado no sistema Xhome é um hardware de mercado, baseado no chip enc28j60 da Microchip (MICROSHIP, 2006). Este chip implementa um controlador Ethernet *stand-alone* com interface SPI. O Shield pode ser visto na figura 17.

Figura 17 – Ethernet Shield para Arduino



Fonte: <http://www.webtronico.com/ethernet-shield-para-arduino.html>

Para a comunicação do microcontrolador ATmega328 com o enc28j60, foi utilizada a biblioteca Ethercard (ETHERCARD, 2013), que se encontra disponível através da licença para softwares livres GPL2. Este driver foi concebido para ser compatível com o Arduino IDE, apesar de não ser um código maduro, necessitando atualmente de revisões e correções de bugs.

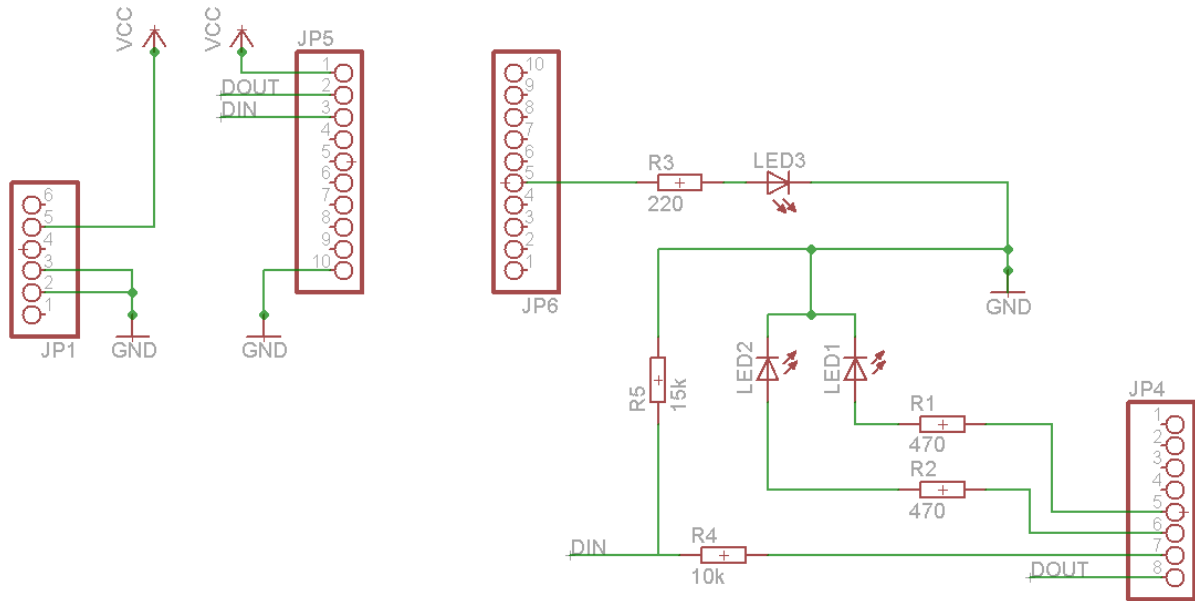
De qualquer forma, para o sistema desenvolvido, as funcionalidades disponibilizadas atendem as necessidades de maneira satisfatória. A biblioteca, um fórum de discussões e exemplos de implementações podem ser encontrados no site que mantém o Ethercard.

### 3.3.3 Placa de interface Arduino/Xbee

A placa de interface desenvolvida para o Gateway do sistema Xhome é uma placa de roteamento dos pinos de comunicação USART do Arduino e de indicação de status do dispositivo, através de LEDs.

O esquemático é apresentado na figura 18.

Figura 18 – Esquemático interface Arduino/Xbee

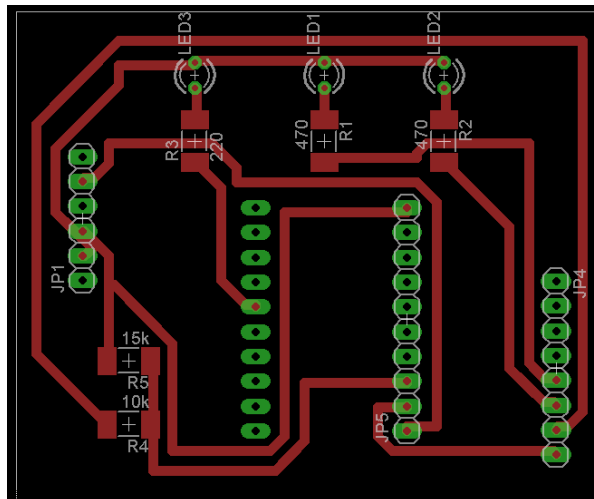


Fonte: Elaborada pelo autor

O arduino funciona em 5V, ao passo que o módulo Xbee opera em 3.3V. Por este motivo, um divisor de tensão, formado por um resistor de 15kΩ e um resistor de 10kΩ, é utilizado para a ligação das interfaces USART do ATmega328 e do Xbee. Por outro lado, o kit de desenvolvimento Arduino UNO oferece, através de um regulador interno de tensão, uma saída de 3.3V. Esta saída do Arduino está sendo utilizada para alimentar o módulo Xbee. Além disto, outros três pinos são utilizados para ligar LEDs que indicarão estado de operação do sistema.

A figura 19 apresenta o layout que foi derivado do esquemático acima.

Figura 19 – Layout interface Arduino/Xbee



Fonte: Elaborada pelo autor

O módulo Xbee é conectado no centro da placa, através de um soquete de espaçamento padrão de 2,54mm. Na realidade, o módulo Xbee possui pinos com um distanciamento não usual de 1mm. Por este motivo, uma placa de interface está sendo utilizada para fazer a conversão das dimensões mecânicas, possibilitando a conexão do módulo.

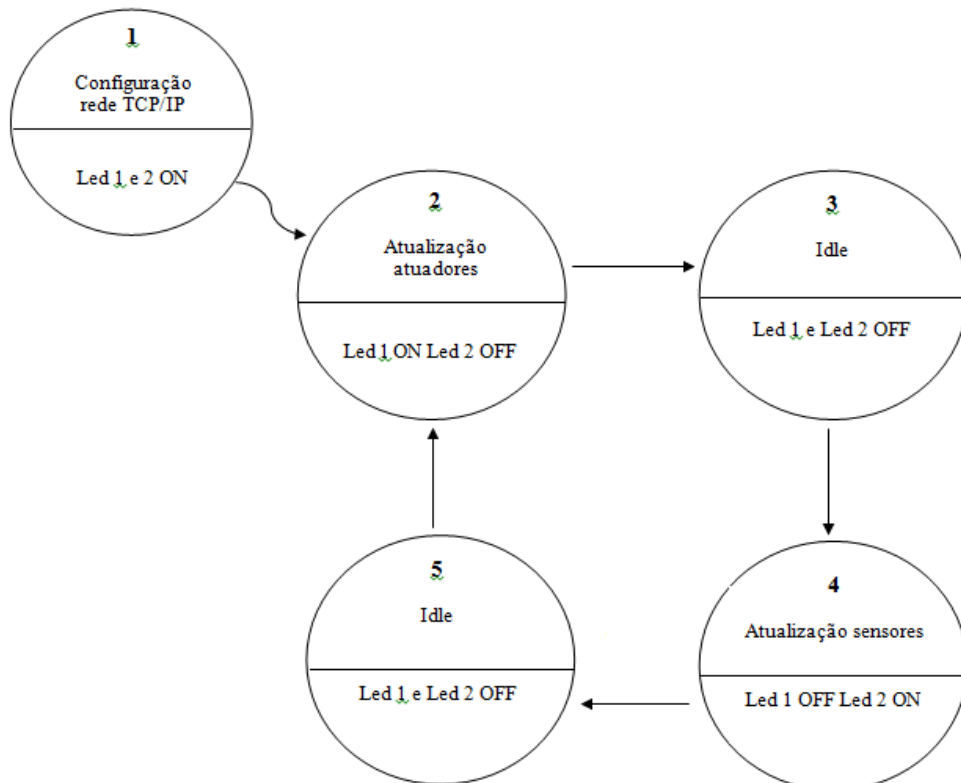
Os JP1 e JP4 indicados no layout são as barras de pinos utilizadas para conexão da placa com o Shield Ethernet, que é ligado logo abaixo da mesma. Por último vemos o divisor de tensão que propicia a conversão das tensões de maneira adequada e os LEDs de indicação do estado do sistema.

### 3.3.4 Programação do microcontrolador

O microcontrolador ATmega328 foi programado através do *SDK* fornecido pelo fabricante, o Arduino IDE versão 1.0.3. A seguir será apresentada uma descrição qualitativa da lógica de programação, conforme foi implementada.

O microcontrolador foi programado para funcionar através do princípio de uma máquina de estados, conforme a figura 20.

Figura 20 – Máquina de estados programada no ATmega328



Fonte: Elaborada pelo autor

No estado 1, o Shield Ethernet está em fase de estabelecimento de uma conexão com o roteador da rede TCP/IP da residência. Neste momento, ele obtém seu endereço IP através do servidor DHCP do roteador e faz uma busca pelo endereço IP do site Xively, através do servidor DNS existente na residência. Para indicar que o Gateway está em fase de configuração, os seus LEDs 1 e 2 ficam acesos. Quando ambos apagam, o sistema entra efetivamente em operação.

A transição do estado 1 para o estado 2 geralmente demora 15s, porém pode variar dependendo da velocidade da internet residencial e até mesmo falhar e caso de indisponibilidade da rede. Salvo esta transição, todas as outras transições de estados são fixas, ocorrendo a cada 500ms.

No estado 2, o microcontrolador busca no site Xively o valor para os atuadores 1 e 2, que estão localizados nas placas de monitoramento/atuação 1 e 2 respectivamente. Após receber a resposta do site e filtrar apenas a informação necessária, o microcontrolador *seta* os valores das saídas digitais remotas dos atuadores. Isto é feito através do envio de um pacote de comando remoto AT na rede, cujo número de frame é 0x17h. Enquanto estiver neste estado, o LED 1 fica aceso (LED verde), indicando que os atuadores estão sendo atualizados.

O estado 3 apenas apaga o LED 1, sinalizando que os atuadores já foram atualizados. Neste momento, o microcontrolador está em modo *idle*, sem processar informações relativas ao sistema de automação. Por outro lado, ele trata alguns pacotes que eventualmente possam ter sido recebidos sobre a rede TCP/IP através do chip enc28j60.

No estado 4, o sistema lê as informações dos sensores que foram recebidas pelo módulo Xbee e que agora estão armazenadas no *buffer* de recepção da porta USART do ATmega328. Em seguida, trata estas informações, filtrando apenas os dados necessários para o sistema, e envia através do Shield Ethernet o valor dos sensores para o site Xively. Neste estado, o LED 2 (cor branca) é aceso, indicando que os sensores estão sendo atualizados no site de controle do sistema de automação.

Por fim, o estado 5 segue a mesma lógica de operação do estado 3, apagando os LEDs, indicando o fim do processamento das informações no estado passado e tratando os pacotes TCP/IP que por ventura tenham sido recebidos através da rede de internet da residência.

De maneira simples e qualitativa, este é o funcionamento do sistema, que



opera em um ciclo infinito entre o estados 2 e 5. Durante a implementação do sistema, varias dificuldades técnicas foram encontradas, como é o caso de achar um *driver* adequado para o chip enc28j60 e contornar as suas limitas/*bugs*. Além disto, o sistema lida com uma diversidade grande de protocolos (Zigbee, HTTP, USART, SPI) e trabalha com limitações de utilização da memória RAM, adicionando ainda outras dificuldades técnicas na sua implementação. Para maiores detalhes sobre o código desenvolvido, ele está disponível de maneira integral no Anexo A.

### 3.4 ATUADORES E SENSORES

As placas de monitoramento/atuação constituem os nós remotos da rede de automação residencial, em oposição ao Gateway Ethernet/Zigbee, que coordena a rede. Em uma aplicação real do sistema Xhome, cada placa seria instalada em um cômodo da residência e seria responsável por controlar os eletrodomésticos que nele se encontram.

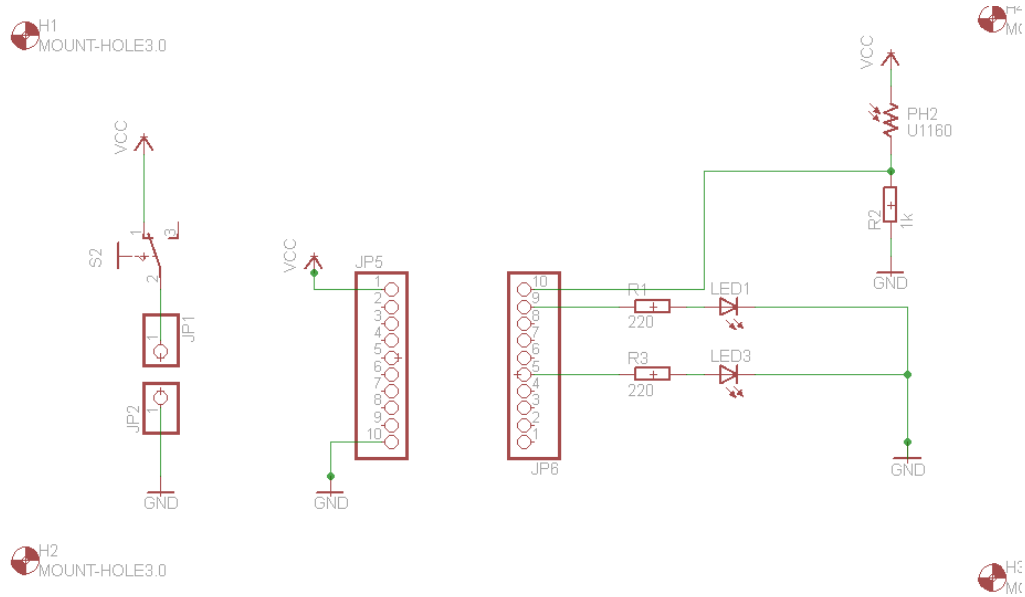
No escopo deste projeto de desenvolvimento de protótipo, foram desenvolvidas duas placas de monitoramento/atuação. Elas são muito similares, apenas apresentando diferença ao nível dos sensores utilizados. Em uma das placas, foi utilizado um botão simples. Ao passo que na outra foi utilizado um LDR, permitindo medir a intensidade luminosa de um determinado ponto da residência.

Para fins de análise, dada a similaridade dos projetos de hardware, apenas a placa com o LDR será apresentada. O esquemático desta placa pode ser visto na figura 21.

Em termos de alimentação, a placa está ligada em duas pilhas AA alcalinas, de 1,5V através de um interruptor que permite ligar/desligar o sistema. O Xbee é conectado à placa através de uma outra placa de interface que converte o espaçamento dos pinos de 1mm para 2.54mm, mesma solução utilizada no projeto de hardware do gateway Ethernet/Zigbee.

Nestas placas, são utilizados dois LEDs. O LED 1 representa o atuador, que na prática pode ser substituído por um relé e o correspondente circuito transistorizado para acionamento de cargas (lâmpadas, ar-condicionado, etc). O segundo LED serve para indicar o status do módulo Xbee, que quando em operação normal, deve piscar com uma frequência fixa em torno de 2Hz.

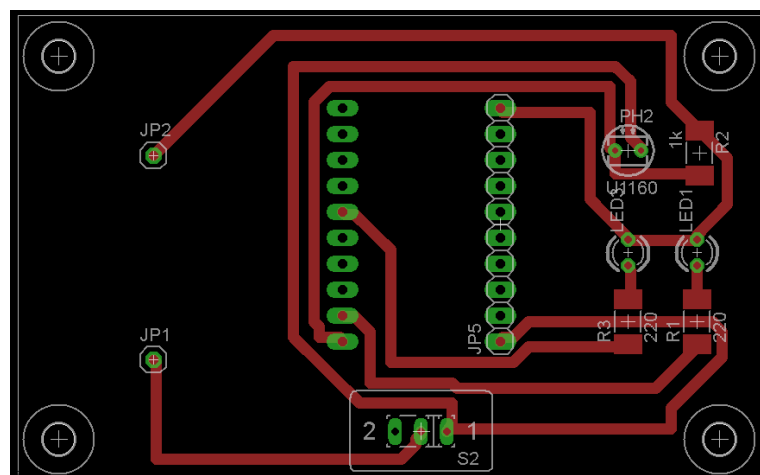
Figura 21 – Esquemático placa monitoramento/atuação



Fonte: Elaborada pelo autor

O LDR está conectado de maneira a formar um circuito divisor de tensão com um resistor de 1KΩ. O valor do resistor foi escolhido de forma a maximizar a excursão de sinal na entrada do ADC dadas as condições normais de luminosidade ambiente em residências.

A figura 22 apresenta o *layout* que foi obtido a partir do esquemático acima referido.

Figura 22 – *Layout* placa monitoramento/atuação

Fonte: Elaborada pelo autor

O modulo Xbee está posicionado no centro da placa, as pilhas do lado esquerdo (onde estão previstas as conexões dos suportes) e os outros componentes do lado direito do *layout*. Adicionalmente, foram previstos espaços para inserção de

apoios mecânicos na placa, de forma a permitir que se acomode horizontalmente sobre um plano, como uma mesa.

### 3.5 XIVELY

Xively é um site que tem por objetivo promover o desenvolvimento do conceito *Internet of Things*, através da disponibilização de forma gratuita de uma plataforma de serviços on-line. Conforme as palavras da empresa criadora do site (XIVELY,2013):

*The Internet of Things promises to revolutionize the relationship between people and their world, but bringing transformative connected products to market takes more than a great idea. It requires tools, infrastructure and know-how that most businesses don't have the time, money or capacity to obtain. However, we have a solution! **Xively is a Platform as a Service** that provides everything you need to **simplify and accelerate** the creation of compelling **connected products and solutions**. With Xively, you can focus on innovation instead of infrastructure.*

Entre outras funcionalidades, Xively fornece uma base de dados *on-line* que permite usuários armazenarem e compartilharem informações de sensores instalados ao redor do mundo. O site permite o upload seguro e gratuito destas informações, sendo possível escolher se um sensor é público, podendo ser acessado por outros usuários, ou é de caráter privado, por exemplo. Além disso, parâmetros como a localização geográfica do sensor e comentários podem ser associados ao mesmo. A página inicial do site pode ser vista na figura 23.

Figura 23 – Página inicial do site Xively.com



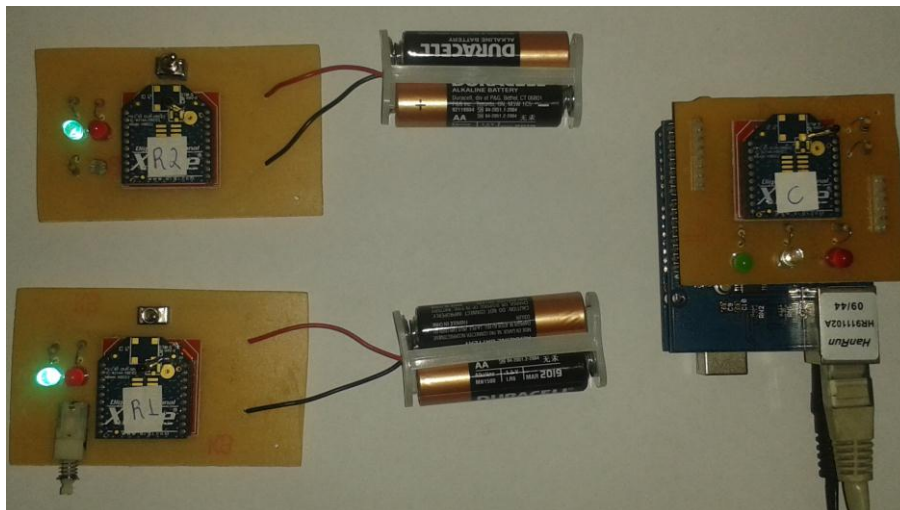
Fonte: <https://xively.com/>

Outras facilidades oferecidas por este site são a comunicação facilitada para envio e recebimento de dados de sensores, utilizando uma API disponibilizada e a geração automática de gráficos históricos. Além disto, o site Xively permite a configuração de *triggers* associados aos *data streams* (nomenclatura utilizada para denominar os pontos de sensores/atuadores). Desta forma, por exemplo, quando um sensor que controla uma porta de entrada de uma residência for acionado, podemos especificar que queremos receber uma mensagem via *Twitter* nos alertando. Esta é uma forma simples de utilizar redes sociais para monitoramento da residência, evitando acessar uma ferramenta específica para tal. Por último, ele permite a extração dos dados utilizando diversos formatos (XML, JSON, CSV) e a conectividade com outros sites é facilitada.

### 3.6 ANALISE DA PERFORMANCE DO SISTEMA

O sistema final montado é apresentado na figura 24.

Figura 24 – Hardwares desenvolvidos para o Xhome



Fonte: Elaborada pelo autor

À esquerda estão localizadas as duas placas de monitoramento/atuação desenvolvidas, uma contendo um botão como sensor e a outra contendo um LDR, possibilitando aferir a intensidade luminosa em um determinado local. Os Xbees estão sinalizados como R1 e R2, de forma a poder distinguir onde estão localizados os atuadores 1 e 2 e o sensores 1 e 2 respectivamente. Em uma aplicação dentro de um sistema de automação residencial, essas placas estariam localizadas junto aos

eletrodomésticos ou dispositivos eletrônicos controlados. Um exemplo seria instalar as placas junto a uma lâmpada ou ao interruptor que a aciona.

À direita está localizado o *gateway* Ethernet/Zigbee, o qual coordena a rede. Como explicado anteriormente, este dispositivo é responsável por alinhar o valor dos atuadores com os que estão registrados na base de dados on-line e enviar os novos valores dos sensores, recebidos via wireless, ao Xively.

A interface gráfica do sistema, hospedada no site Xively, é apresentada na figura 25. Nela temos o valor dos atuadores e dos sensores para ambos os nós da rede Zigbee. Um número 1 (um) no atuador ativa o LED, representando o acionamento do eletrodoméstico, ao passo que um 0 (zero) desativa o atuador. Os sensores, por estarem ligados em ADCs de 10 bits, possuem valores que estão compreendidos em uma escala de 0 à 1023.

Figura 25 – Interface sistema Xhome

The screenshot displays the Xively system interface for a device named 'Sistema Automação - TCC'. The interface is divided into several sections:

- Device Information:** Shows the device is 'Activated' at 25-05-2013 22:45:03. It includes a 'Deploy' button and lists the Feed ID (1655821478), Feed URL (https://xively.com/feeds/1655821478), and API Endpoint (https://api.xively.com/v2/feeds/1655821478).
- Private Device Details:** Lists Product ID, Product Secret, Serial Number, and Activation Code.
- Channels:** A table showing the status of actuators and sensors.
 

Channel	Value
Atuador1	1
Atuador2	1
Sensor1	1023
Sensor2	430
- Request Log:** A log of HTTP requests, showing a sequence of PUT and GET requests to the 'feed' endpoint at 19:02:33-0300.
- API Keys:** A section for managing API keys.

Fonte: <https://xively.com/>

À direita, em *Request Log*, podemos visualizar todas as requisições recebidas pelo site Xively de agentes externos. Neste caso, pode-se observar as requisições HTTPs de GET e PUT enviadas periodicamente pelo *gateway* de forma a atualizar os atuadores e os sensores da rede respectivamente.

### 3.6.1 Análise do tempo de resposta do sistema

De uma maneira geral, o sistema apresentou um desempenho satisfatório. Os atuadores e sensores são atualizados conforme esperado e em tempo satisfatório. O

sistema ficou operando por um dia completo sem apresentar problemas ou interrupções, se demonstrando estável.

Analisando mais profundamente o funcionamento do sistema, pode-se concluir que o tempo para atualização dos sensores e dos atuadores é diferente. A seguir é mostrada uma análise, baseada em hipóteses simplificadoras, que buscar medir e compreender estas diferenças.

### 3.6.1.1 Análise do tempo de atualização dos atuadores

Para a atualização dos atuadores, o tempo total do fluxo da informação pode ser escrito como segue:

$$T_{\text{total/atuadores}} = T_{\text{interface/site}} + T_{\text{site/pooling}} + T_{\text{pooling/recebimento}} + T_{\text{gateway/atuador}} \quad (1)$$

Em um primeiro instante, devemos contabilizar o tempo entre a requisição de alteração do valor do usuário na interface (*smartphone*, *tablet*, etc), o tempo de transito desta informação até o servidor que hospeda o site e a atualização da sua base de dados. Este primeiro tempo foi chamado de  $T_{\text{interface/site}}$ .

Como o microcontrolador está programado para funcionar através de pooling, ou seja, requisitar com uma determinada periodicidade os valores dos atuadores, existirá outro atraso relativo ao não sincronismo da atualização da base de dados e a requisição do novo valor. No pior dos casos, este tempo será de 2s, tempo completo para completar um ciclo da máquina de estados. À este tempo, foi dado o nome de  $T_{\text{site/pooling}}$ .

Em um terceiro momento, o Gateway irá efetivamente requisitar o novo valor do atuador. Este processo envolve enviar uma requisição GET HTTP, através de pacotes TCP/IP ao servidor, o conseqüente processamento do servidor e o envio e recebimento dos valores pelo Gateway novamente. À este tempo, foi dado o nome de  $T_{\text{pooling/recebimento}}$ .

Por último, entre o recebimento do pacote com os novos valores dos atuadores, o processamento da informação e o envio da ordem de atualização do atuador na placa de monitoramento/atuação, transcorre um tempo denominado  $T_{\text{gateway/atuador}}$ . Na prática este tempo pode ser ignorado, por ser muito inferior aos outros tempos em questão. De fato, o envio e recepção de pacotes na rede interna de automação residência Zigbee se faz praticamente de maneira instantânea.

Para medir os tempos  $T_{\text{interface/site}}$  e  $T_{\text{pooling/recebimento}}$  um experimento foi elaborado. Dois computadores independentes foram conectados ao site Xively, no

mesmo *feed*. Dentro deste *feed*, o valor de um atuador foi modificado pelo computador 1, e se aguardou a modificação no computador 2. O setup do experimento é mostrado na figura 26.

Figura 26 – Experimento para medição do tempo de propagação



Fonte: Elaborada pelo autor

Após 10 medidas nas mesmas condições, chegou-se a uma média de tempo total do percurso de 0,97s. Por questão de simplificação e considerando as imprecisões de medição, visto que ela foi feita de forma manual, vamos tomar como 1s a estimativa para o percurso total da informação e 0,5s o tempo para metade do caminho. Com estas hipóteses, chegamos aos seguintes números aproximados:

$$T_{\text{interface/site}} = 0,5\text{s} \quad T_{\text{pooling/recebimento}} = 1\text{s}$$

Desprezando o tempo  $T_{\text{gateway/atuador}}$ , pelas razões acima expostas e considerando  $T_{\text{site/pooling}}$  com os valores 0s e 2s para respectivamente o melhor e o pior caso, através de (1) obtemos os seguintes valores para  $T_{\text{total/atuadores}}$ :

$$T_{\text{total/atuadores (melhor caso)}} = 1,5\text{s}$$

$$T_{\text{total/atuadores (pior caso)}} = 3,5\text{s}$$

Os resultados das medições para o tempo de atualização dos atuadores são mostrados na tabela 2.

Tabela 2 – Medições tempo de atualização atuadores

	1	2	3	4	5	6	7	8	9	10	Média
Atuador (s)	1,96	2,48	2,1	2,48	2,29	2,82	2,44	2,45	2,93	2,68	2,463

Fonte: Elaborada pelo autor

Percebe-se que, apesar das hipóteses simplificadoras que foram feitas, as amostras respeitam relativamente bem os resultados do equacionamento. De fato, os valores mínimos e máximos foram respeitados. Além disto, a média dos valores empíricos fica muito próxima da média entre o melhor e o pior caso, calculados analiticamente.

Através desta análise, pode-se afirmar que o tempo esperado (mais provável)

para atualização de um sensor neste sistema, dadas as mesmas condições de configuração do sistema e considerada a mesma performance do site Xively, é de 2,5s.

### 3.6.1.2 Análise do tempo de atualização dos sensores

Para a atualização dos sensores, o tempo total do fluxo da informação pode ser escrito como segue:

$$T_{\text{total/sensores}} = T_{\text{sensor/sample}} + T_{\text{sample/pooling}} + T_{\text{pooling/site}} + T_{\text{site/interface}} \quad (2)$$

O primeiro tempo apresentado na equação,  $T_{\text{sensor/sample}}$ , representa a diferença temporal entre a mudança do valor do sensor e a efetiva tomada da amostra pelo ADC do Xbee. As amostras foram programadas para serem feitas a cada 1,5s. Desta forma, pode-se considerar o melhor caso como 0s e o pior como 1,5s.

O tempo  $T_{\text{sample/pooling}}$  representa a diferença temporal entre o fato de a amostra ter sido enviada ao Gateway e ela ter sido processada. Como já foi citado, o Gateway funciona em sistema de *pooling*, rodando uma máquina de estados com periodicidade de 2s. Logo, podemos inferir o melhor caso em 0s e o pior caso para este tempo como sendo 2s.

Por último, os tempos  $T_{\text{pooling/site}}$  e  $T_{\text{site/interface}}$  representam as diferenças temporais entre a saída da informação do sensor do Gateway até o seu armazenamento na base de dados do Xively e o trânsito dessa informação até a interface do usuário respectivamente. Conforme hipótese mencionada anteriormente, vamos considerar cada trajeto como sendo aproximadamente 0,5s, totalizando uma soma de 1s.

Levando em conta as hipóteses supracitadas e os tempos medidos, chegamos aos valores para  $T_{\text{total/sensores}}$ :

$$T_{\text{total/sensores}} \text{ (melhor caso)} = 1\text{s}$$

$$T_{\text{total/sensores}} \text{ (pior caso)} = 4,5\text{s}$$

Os resultados das medições para o tempo de atualização dos sensores são apresentados na tabela 3.

Tabela 3 – Medições tempo de atualização dos sensores

	1	2	3	4	5	6	7	8	9	10	Média
<b>Sensor (s)</b>	4,31	2,81	3,95	1,95	3,1	3,57	3,25	3,7	3,3	3,71	<b>3,365</b>

Fonte: Elaborada pelo autor



Nota-se que os limites inferiores e superiores foram respeitados, porém a média dos valores medidos desta vez se apresentou mais distante da média entre o melhor e o pior caso. De qualquer forma, pode-se afirmar, baseando-se nos valores medidos, que o tempo esperado para a atualização do valor dos atuadores é de 3,36s.

Percebe-se que este tempo é superior ao tempo de atualização dos atuadores. De fato, esta diferença se explica pois no processo de atualização do valor dos sensores as amostras são tomadas com uma periodicidade de 1,5s, de maneira discreta. A soma deste valor com o tempo de *pooling* do microprocessador ATmega gera um atraso médio maior em comparação ao processo de atualização dos atuadores.

### 3.6.2 Outros parâmetros

No início deste capítulo, foram apresentados os requisitos que o sistema deveria atender, são eles:

- Apresentar uma interface com a internet, através da qual seja possível atuar e monitorar dispositivos da rede de automação residencial
- Escalável
- Fácil instalação
- Baixo consumo de energia
- Baixo custo

O sistema possui uma interface *Web*, que pode ser acessada através de qualquer dispositivo com acesso à internet, além de permitir o monitoramento de sensores e a atuação sobre dispositivos.

O sistema é escalável. De fato, neste trabalho foram utilizadas apenas duas placas de monitoramento/atuação, apresentando esquemáticos muito similares. Caso deseje-se automatizar mais dispositivos de uma residência, basta replicar o projeto destas placas. Em outras palavras, o crescimento do sistema é viável e se dá de maneira simples. Além disto, o sistema pode ser instalado de maneira simples, por ser baseado em tecnologia *wireless*.

O consumo de corrente médio medido das placas de atuação/monitoramento é de 40mA, sobre 3.3V. A maior parte desta corrente está sendo destinada aos LEDs de indicação, que podem ser omitidos em aplicações reais de automação

residencial. Além disto, se programados em modo *End Device*, os módulos Xbee podem ser configurados em modo Sleep, consumindo apenas alguns microampères e, portanto, tendo vida útil de mais de um ano quando ligados em baterias.

Por último, estima-se que o custo total dos dispositivos utilizados neste projeto é de 500 reais, sendo a maior parte empregada nos módulos Xbee. O preço médio de um módulo Xbee utilizado está em torno de 75 reais. Um projeto similar utilizando tecnologia de mercado não sairia por menos de 2 mil reais, ao passo que sistemas completos de automação residencial de mercado custo entre 10mil e 30mil reais (G1 O portal de notícias da Globo, 2013).

A grande desvantagem do sistema como concebido atualmente é a forte dependência da internet como meio de comunicação. De fato, caso o serviço esteja indisponível dentro da residência, o Xhome não pode ser acessado. Uma maneira de contornar este problema seria a configuração de interfaces de controle dentro da residência que se comuniquem diretamente com o coordenador da rede, via protocolo Zigbee.

Uma outra solução é o desenvolvimento de um *gateway* Zigbee/Wifi, que poderá ser acessado por *tablets*, *laptops* ou *smartphones* quando estes estiverem nas redondezas da residência. Mesmo que o serviço de internet não esteja disponível, a rede Wifi interna da residência continuará funcionando.

## 4 CONCLUSÃO

Poucos sistemas de automação residencial são concebidos sem fio. No caso de residências em construção, a passagem de cabos não é complicada e pode, em algumas situações, se mostrar mais barata em comparação ao uso de tecnologias *wireless*. Para residências já existentes, porém, a instalação de novos dutos ou simplesmente a passagem dos cabos de comunicação do sistema de automação residencial se torna cara e complicada. Nestes casos, a adoção de sistemas *wireless* apresenta uma série de vantagens.

Este trabalho demonstrou a viabilidade de construção de um sistema de automação residencial *wireless* de baixo custo. De fato, a ideia base para a implantação de um sistema mais amplo foi especificada. Através da replicação das placas de monitoramento/atuação e alguns pequenos ajustes no código de programação do microcontrolador, é possível controlar uma residência completa.

A solução como foi concebida é de fácil instalação em residências, pelos motivos acima expostos. A expansão e manutenção do sistema também são simplificadas, dado que a rede Zigbee se autoconfigura caso algum nó seja adicionado ou retirado. Além disto, os dispositivos de comunicação consomem pouca energia, possibilitando longo período de autonomia quando ligados em baterias. O fato de não necessitarem estar conectados diretamente a cabos de energia também é um diferencial.

Conforme apresentado no início deste trabalho, uma vasta quantidade de soluções e empresas estão presentes atualmente no mercado de automação residencial. Estes sistemas muitas vezes utilizam protocolos proprietários e não possuem *gateways* que permitam a comunicação com soluções de terceiros. Esse panorama não é sustentável, visto que os sistemas podem facilmente se tornar obsoletos. Além disto, uma futura integração de sistemas de automação residencial com *Smart Grids* ou *Smart Cities* está comprometida. Por essas e outras razões, a padronização de um protocolo para automação residencial é essencial.

O protocolo Zigbee vem se tornando largamente utilizado em diversas áreas, como redes de sensores, controle de veículos não tripulados, entre outros. Mesmo alguns sistemas de monitoramento de fazendas e rebanhos já foram desenvolvidos, utilizando-se do módulo Xbee-PRO. Esse módulo, com um custo levemente mais elevado, possibilita a comunicação entre distâncias de até 1,0 Km em campo aberto.

Além da Digi<sup>®</sup>, existem outros fabricantes no mercado que propõem dispositivos compatíveis com o protocolo Zigbee. O fato deste padrão estar sendo empregado em larga escala aumenta as chances de existir interoperabilidade entre o sistema desenvolvido e outros sistemas de mercado, desta forma evitando que ele se torne obsoleto.

Em suma, a solução Xhome é robusta e atende os requisitos que foram impostos no início do desenvolvimento do sistema. Como ela se apresenta hoje, pode ser empregada em uma residência real. Por outro lado, algumas melhorias poderiam ser desenvolvidas. Este é o caso, por exemplo, do desenvolvimento de uma interface de configuração do sistema. Atualmente toda a informação sobre o modo como o sistema deve operar se encontra no código que programa o microcontrolador ATmega328.

Uma segunda melhoria seria o desenvolvimento de uma placa de atuação com saídas em infravermelho, permitindo controlar dispositivos digitais como televisões, DVDs e *Home Theaters*. Uma análise mais avançada sobre os tempos de propagação da informação entre os dispositivos, conforme apresentado no último capítulo deste trabalho, permitiria uma melhoria na velocidade de operação do sistema. Estes pontos ficam aqui citados como sugestão para uma eventual continuação no desenvolvimento do sistema.

## REFERÊNCIAS

REDE GLOBO. **G1 O portal de notícias da Globo**. Disponível em <<http://g1.globo.com/economia/pme/noticia/2013/05/empresa-fatura-r-1-milhao-com-servicos-de-automacao-residencial.html>>. Acesso em: 28 de maio 2013.

WIKIPEDIA, A ENCICLOPEDIA LIVRE. **Modelo V**. Disponível em <[http://pt.wikipedia.org/wiki/Modelo\\_V](http://pt.wikipedia.org/wiki/Modelo_V)>. Acesso em: 15 de dez. 2012.

ROBERTO MURATORI, José; HENRIQUE DAL BO, Paulo. **Automação Residencial: histórico, definição e conceitos**. 2012.

SeeTool – Solutions for KNX. **Application 1.0.0.0.2 Movement dependent automatic light control**. Schneider Electric. 2011.

ABB i-bus® KNX. **Application Manual Lighting**. ABB. 2010.

PRUDENTE, Francesco. **Automação Predial e Residencial: Uma Introdução**. Rio de Janeiro: LTC, 2011. ISBN 978-85-216-0617-8.

WIKIPEDIA, A ENCICLOPEDIA LIVRE. **X10 (industry standard)**. Disponível em <[http://en.wikipedia.org/wiki/X10\\_\(industry\\_standard\)](http://en.wikipedia.org/wiki/X10_(industry_standard))>. Acesso em: 30 de mar. 2013.

LONWORKS - ECHELON. **Requirements for Device Networking**. Disponível em <<http://www.echelon.com/technology/lonworks/networking-criteria.htm>>. Acesso em: 15 de fev. 2013.

XIVELY. **XIVELY – Public Cloud for The Internet of Things**. Disponível em <<https://xively.com/>>. Acesso em: 15 de maio 2013.

WIKIPEDIA, A ENCICLOPEDIA LIVRE. **Zigbee**. Disponível em <<http://pt.wikipedia.org/wiki/ZigBee>>. Acesso em: 30 de abril 2013.

DIGI INTERNATIONAL INC. **Product Manual v1.xEx - 802.15.4 Protocol: For RF Module Part Numbers: XB24-A...-001, XBP24-A...-001**. USA: 2009.

MICROSHIP. **ENC28J60 Datasheet**: Stand-Alone Internet Controller with SPI Interface. Microship Technology Inc. 2006.

ETHERCARD. **jcw/ethercard** - **GitHub**. Disponível em <<https://github.com/jcw/ethercard>>. Acesso em: 7 de fev. 2013.

Arduino. **Arduino Uno**. Disponível em <<http://arduino.cc/en/Main/arduinoBoardUno>>. Acesso em: 17 de maio 2013.

FALUDI, Robert. **Building Wireless Sensor Networks**: A Practical Guide To The Zigbee Mesh Networking Protocol. USA: O'Reilly, 2010. ISBN 978-0-596-80773-3.

## **ANEXO A – Código de programação ATmega328**

```

#include <EtherCard.h>

// ethernet interface mac address, must be unique on the LAN
#define FEED "1655821478"
#define APIKEY "DuARnbQrgVb3O2vneqdEJ622mardm6hmuu6oXSmOpeO3MXmz"

static byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };
static byte snR1[] = { 0x00,0x13,0xA2,0x00, 0x40,0x79,0x61,0x42 };
static byte snR2[] = { 0x00,0x13,0xA2,0x00, 0x40,0x79,0x61,0x45 };

byte Ethernet::buffer[700];
uint32_t timer, stachcount;

char website[] PROGMEM = "api.xively.com";
Stash stash;
int led1 = 2, led2 = 3, ESTADO=0;
int analogValueR1, analogValueR2;

// função chamada ao receber os valores do atuadores do site Xively.com
static void callback_atuadores (byte status, word off, word len) {

    int newValorAtuador1,newValorAtuador2;

    Ethernet::buffer[off+700] = 0;
    newValorAtuador1 = valor_atuador(off, "Atuador1,");
    newValorAtuador2 = valor_atuador(off, "Atuador2,");

    if (newValorAtuador1==1)
    {
        setRemoteState(5,1);
    }
    else if (newValorAtuador1==0)
    {
        setRemoteState(4,1);
    }
};

```



```

if (newValorAtuador2==1)
{
    setRemoteState(5,2);
}
else if (newValorAtuador2==0)
{
    setRemoteState(4,2);
};
}

```

```

int valor_atuador (word pos, char nomeProcurar[]){

```

```

    int ValorAtuador=9999;
    int lengthNome=0;
    while(nomeProcurar[lengthNome])
    lengthNome++;

```

```

while (Ethernet::buffer[pos])
{

```

```

    if( strcmp ((char*)&Ethernet::buffer[pos], nomeProcurar, lengthNome) == 0 ) {
        // Datastream encontrado
        pos += lengthNome; // Pula
        break;
    }
    pos++;
}

```

```

while (Ethernet::buffer[pos])
{

```

```

    if( strcmp ((char*)&Ethernet::buffer[pos], "Z", 2) == 0 ) {
        // Informação encontrada
        pos += 2; // Pula valor
        char ch = Ethernet::buffer[pos++];
        ValorAtuador=0;
        while(ch >= '0' && ch <= '9' ) { // Apenas digitos
            ValorAtuador *= 10;

```

```
        ValorAtuador += (ch - '0');
        ch = Ethernet::buffer[pos++];
    }
    break;
}
pos++;
}

return ValorAtuador;
}

void atualiza_sensores (){

    int length;
    boolean R1=true,R2=true;

    while (Serial.available())
    {
        // Procura o byte de inicio de pacote
        if (Serial.read() == 0x7E) {

            int lengthHigh = Serial.read();
            int lengthLow = Serial.read();
            length = lengthLow + (lengthHigh * 256);

            if (Serial.read() == 0x92) {

                R1=true;
                R2=true;

                for (int i = 0; i<8; i++) {
                    byte buff = Serial.read();

                    if (buff!=snR1[i]) {
                        R1=false;
                    }
                }
            }
        }
    }
}
```

```
    if (buff!=snR2[i]) {
        R2=false;
    }
}

for (int i = 0; i<(length-3-8); i++) {
    byte discard = Serial.read();
}

int analogHigh = Serial.read();
int analogLow = Serial.read();
int analogValue = analogLow + (analogHigh * 256);

if (R1) {
    analogValueR1=analogValue;
}

if (R2) {
    analogValueR2=analogValue;
}
// Discarta checksum
byte discard = Serial.read();
}
}
}

stachcount = Stash::freeCount();
byte sd = stash.create();

if (analogValueR1<=1023) {
    stash.print("Sensor1,");
    stash.println((word) analogValueR1);
}

if (analogValueR2<=1023) {
    stash.print("Sensor2,");
    stash.println((word) analogValueR2);
}
```

```

stash.save();

Stash::prepare(PSTR("PUT http://$F/v2/feeds/$F.csv HTTP/1.0" "\r\n"
    "Host: $F" "\r\n"
    "X-PachubeApiKey: $F" "\r\n"
    "Content-Length: $D" "\r\n"
    "\r\n"
    "$H"),
    website, PSTR(FEED), website, PSTR(APIKEY), stash.size(), sd);
// Envia o pacote
ether.tcpSend();
}

void atualiza_atuadores (){
    ether.browseUrl(PSTR("/v2/feeds/"FEED".csv?key="APIKEY), "", website,
callback_atuadores);
}

void setRemoteState(int value, int nRemote) {

Serial.write(0x7E); // start byte
Serial.write(0x0); // high part de length (sempre zero)
Serial.write(0x10); // low part de length (numero de bytes que seguem, sem checksum)
Serial.write(0x17); // 0x17 é remoto AT command
Serial.write(0x0); // frame id setado zero para no reply
// ID do destina, ou 0xFFFF para broadcast
for (int i = 0; i<8; i++) {

    if (nRemote==1) {
        Serial.write(snR1[i]);
    }

    if (nRemote==2) {
        Serial.write(snR2[i]);
    }
}
}

```

```

}

// 16 bit do destino, ou 0xFFFE se desconhecido
Serial.write(0xFF);
Serial.write(0xFE);
Serial.write(0x02); // 0x02 to apply changes immediately on remote
// comando com caracteres ASCII - Aciona LED
Serial.write('D');
Serial.write('1');
Serial.write(value);
// checksum
long sum = 0x17 + 0xFF + 0xFE + 0x02 + 'D' + '1' + value;

for (int i = 0; i<8; i++) {

    if (nRemote==1) {
        sum += snR1[i];
    }

    if (nRemote==2) {
        sum += snR2[i];
    }
}

Serial.write( 0xFF - ( sum & 0xFF) );
}

void setup () {

    ESTADO=1;
    Serial.begin(9600);
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    ether.begin(sizeof Ethernet::buffer, mymac, 10);
    ether.dhcpSetup();
}

```

```
    ether.dnsLookup(website);
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    ESTADO=2;
}

void loop () {

    ether.packetLoop(ether.packetReceive());
    if (millis() > timer) {
        timer = millis() + 500;

        if (stash.freeCount() <= 25) {
            Stash::initMap(56);
        }

        if (ESTADO==2){
            digitalWrite(led1, HIGH);
            ESTADO=3;
            atualiza_atuadores();}
        else if (ESTADO==3){
            digitalWrite(led1, LOW);
            ESTADO=4;
        }
        else if (ESTADO==4){
            digitalWrite(led2, HIGH);
            ESTADO=5;
            atualiza_sensores();
        }
        else if (ESTADO==5){
            digitalWrite(led2, LOW);
            ESTADO=2;
        }
    }
}
```