

# Domótica: Controle de Automação Residencial Utilizando Celulares com Bluetooth

**Eduardo Rodrigues de Abreu**  
abreu@univali.br  
UNIVALI

**Paulo Roberto Oliveira Valim**  
pvalim@univali.br  
UNIVALI

**Resumo:** As empresas de tecnologia enfrentam um paradoxo na disputa pela casa do futuro. De um lado, a automação residencial requer sistemas capazes de se comunicar, como, por exemplo, aparelhos celulares que possam programar a iluminação da residência ou o ar-condicionado e geladeiras ou que transmitam pedidos de compra para o supermercado mais próximo. Do outro lado, entretanto, cada empresa busca impor seu próprio padrão tecnológico aos demais concorrentes, o que apenas atrapalha a integração dos produtos e serviços. Diante deste cenário de mercado, esse trabalho contempla o desenvolvimento de um módulo conversor de protocolo. O sinal será emitido por um celular através da tecnologia Bluetooth o qual irá se comunicar com o módulo “conversor”. Esse fará a adaptação desse sinal a um protocolo legado utilizado na automação residencial. Para demonstração da solução proposta o protocolo ao qual o módulo estará convertendo será o padrão X-10.

**Palavras Chave:** Domótica - Automação - Bluetooth - X-10 - Smart Build

## 1. INTRODUÇÃO

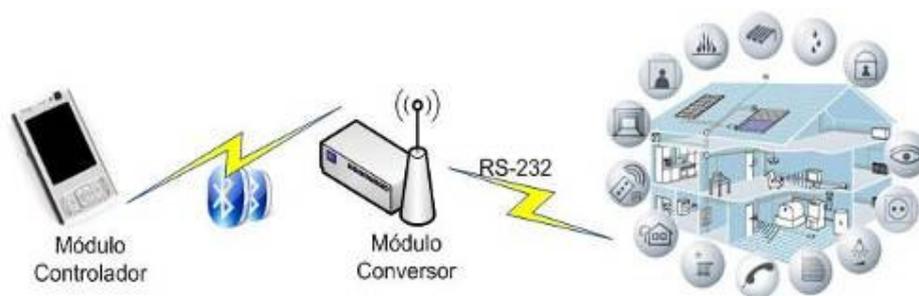
Segundo Roque (2008), domótica é a utilização simultânea da eletricidade, da eletrônica e das tecnologias da informação no ambiente residencial, permitindo realizar a sua gestão, local ou remota, e oferecer um vasta gama de aplicações integradas nas áreas da segurança, conforto, comunicação e gestão de energia.

Entretanto, a realidade exposta pela AURESIDE em 2010 (Associação Brasileira de Automação Residencial) em relação a Roque (2008) é controversa. Hoje o mercado voltado a esse campo da tecnologia passa por grandes desafios dos desenvolvedores, os quais possuem como principal problema a dificuldade ou impossibilidade de integração de todas as funções exercidas pela automação presente dentro de uma residência.

Esse fato se deve grande diversidade de tecnologias e protocolos de comunicação, que por sua vez é resultado de uma falta de padronização. Desse modo, é habitual encontrar dentro de uma propriedade automatizada diversos sistemas os quais não “conversam” entre si, ou seja, fogem do conceito de domótica.

Ciente desse cenário atual, a solução proposta consiste em gerenciar uma residência através de uma interface para celular utilizando a tecnologia *Bluetooth* disponível no mesmo. O celular comunica-se com um microcontrolador que é responsável pelo controle das variáveis presentes na automação residencial.

A solução proposta divide-se em dois módulos: “controlador” e “conversor”. O módulo controlador (software) possui a funcionalidade de realizar a comunicação do celular através de *Bluetooth* com o microcontrolador. Já o módulo “conversor” (*firmware*) realiza a conversão dos comandos enviados pelo celular para o protocolo X-10. Desse modo é possível haver comunicação do microcontrolador com objetos presentes em uma residência os quais já operam sobre esse protocolo. Tendo como realidade das empresas que atuam no mercado de automação residencial a utilização de soluções proprietárias (AURESIDE, 2006), diante a essa restrição, nesse trabalho é apresentado uma interface que possa ser utilizada para qualquer solução mediante uma adaptação no módulo “conversor”. Na Figura 1 é apresentado o diagrama de visão geral do projeto.



**Figura 1:** Celular realizando comunicação com objetos da residência.



## 2.1. CLASSIFICAÇÃO DA DOMÓTICA

Toda proposta de implantação da domótica executa uma série de funções que podem estar integradas ou não a outras funções de outros dispositivos envolvidos no sistema. Segundo Muratori (2004), o modo como ocorre à execução dessas funções define a classificação da domótica em sistemas: autônomos, integrados ou complexos:

- Autônomos: possuem funcionalidades somente de acionar ou desligar dispositivo ou subsistemas. Nessa classificação nenhum dispositivo ou subsistema tem relação um com o outro (TERUEL, 2008, p.28);
- Integrados: possuem múltiplos subsistemas integrados a um único controlador. A limitação deste sistema está no fato que cada subsistema deve ainda funcionar unicamente na forma a qual seu fabricante pretendia. Basicamente trata-se apenas de controle remoto estendido a diferentes locais (TEZA, 2002, p.32). O processamento pode ser centralizado na central de automação ou distribuído pela rede (TERUEL, 2008, p.28);
- Complexos: possuem como grande diferencial a possibilidade da personalização de produtos manufaturados de modo que atenda as necessidades do proprietário. É dependente de comunicação de mão dupla e realimentação de status entre todos os subsistemas (TEZA, 2002, p.33).

## 2.2. FORMAS DE CONTROLE

De acordo com a forma de controle, ou seja, as formas como os diferentes elementos do sistema de controle estão distribuídos dentro da arquitetura do sistema, define-se a forma de controle como sendo centralizada ou descentralizada (ALIEVI, 2008, p. 17).

Utilizando-se a arquitetura centralizada, o controlador centralizado envia as informações para os atuadores e interfaces. Já o recebimento das informações se torna possível devido à presença de sensores, sistemas interconectados e as ações do usuário (CASADOMO, 2008). A arquitetura centralizada traz como principal benefício o fato de a implantação do sistema possuir um custo reduzido comparando-a com a arquitetura descentralizada, entretanto, traz consigo também a desvantagem da grande quantidade de cabeamento, a complexidade do interfaceamento homem-máquina de modo que não corresponde a filosofia dos sistemas domóticos. Muitos sistemas são executados com CLPs (Controlador Lógico Programável), outros sistemas não possuem *bus* de comunicação e inexistência de comunicação entre equipamento (DÍAZ, PARDO, PULIDO, 2001).

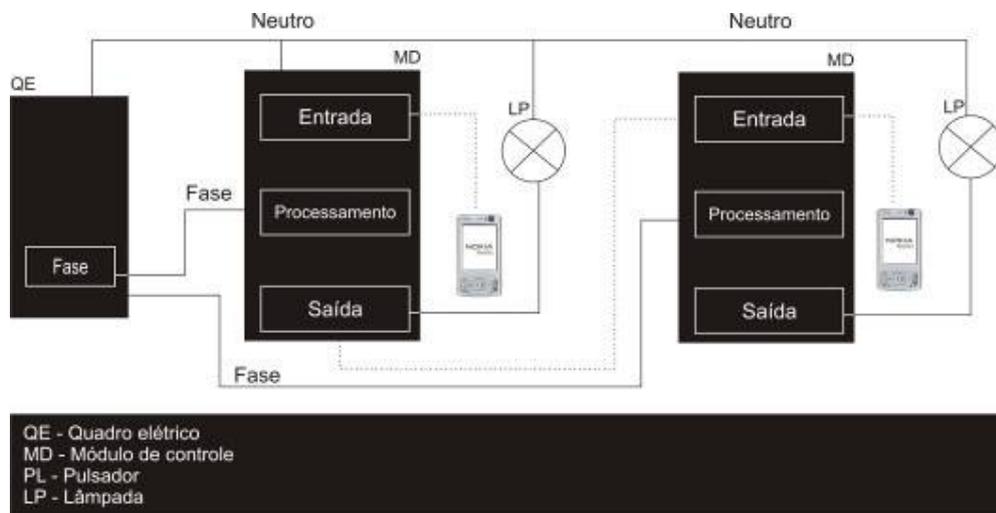
Na arquitetura descentralizada existem diversos controladores interconectados por um *bus* que possibilitam o envio de informações entre eles. Já os atuadores, as interfaces e os sensores não necessariamente comunicam-se com mais de um controlador diretamente, ou seja, a proposta é dividir o sistema para suprir necessidades complexas (CASADOMO, 2008). Esse modelo de arquitetura tem como benefícios tornar os sistemas mais robustos a falhas, fácil desenho das instalações, grande facilidade de uso, ou seja, cumpre todos os requisitos que um sistema domótico deve ter. Entretanto, dependendo do sistema empregado pode haver uma má relação do ponto

controlado em relação ao seu preço. Custo de integração mais elevado em função das tecnologias empregadas. (DÍAZ, PARDO, PULIDO, 2001).

### 2.3. MEIOS DE COMUNICAÇÃO

Quanto à forma de comunicação entre os dispositivos, às soluções propostas pela domótica geralmente são cabeadas ou por radiofrequência. Dentro das soluções cabeadas, podem ser utilizados cabos de dados ou a rede elétrica (TERUEL, 2008, p. 29). Devido ao fato de todas as soluções serem ligadas ao quadro elétrico, não há como ter uma implantação da domótica cem por cento *wireless*.

Nos sistemas com transmissão de dados por radiofrequência, o sistema de ligação é muito parecido com o sistema que usa cabeamento de dados. Tendo como principal diferença a presença de pequenos módulos de automação que são embutidos nas paredes, aos quais, é necessária a ligação do neutro como mostrado na Figura 3 (TERUEL, 2008, p. 33).



**Figura 3:** Controle de lâmpadas por rádio frequência

Fonte: Adaptado de Teruel (2008)

### 2.4. PROTOCOLO E PADRÕES DE COMUNICAÇÃO

Todas as soluções que serão citadas e comentadas a seguir utilizam uma linguagem própria entendida pelo dispositivo que envia e pelo dispositivo que recebe os sinais na comunicação. Essa linguagem estabelece regras de comunicação especificando como e quais termos serão reconhecidos, como os erros serão informados e como serão as transações de conversação. Essa linguagem é o protocolo de comunicação (TERUEL, 2008, p. 35). De acordo com a divisão planejada das funções domóticas (Figura 2) apresentada por Bolzani (2007), os principais protocolos utilizados para controle em sistemas domóticos são: CeBus (*Consumer Electronic Bus*), Z-Wave, X-10, ZigBee, BacNet (*Building Automation and Control Networks*) e LonWorks (*Local Operation Networks*) (Figura 10). Já para dados são: *Fast Ethernet*, *Gigabit Ethernet*, *Power Line*, *HomePNA*, *Wi-Fi* e *Bluetooth*.

Conforme citado na introdução, a aplicação que será apresentada na seqüência utiliza para a parte de controle o protocolo X-10 e para transmissão de dados o *Bluetooth*. O protocolo X-10 foi escolhido de modo a exemplificar, ou seja, para operar sobre outros protocolos deverá ser realizado um ajuste na aplicação.

#### 2.4.1. X-10

O X-10 transmite dados binários através da corrente elétrica utilizando um pulso de sinal na frequência de 60Hz AC (Corrente Alternada), a definição do valor lógico se dá através do cruzamento do sinal pelo ponto zero da curva de frequência (TEZA, 2002, p. 69). Com a intenção de reduzir erros, são utilizados dois cruzamentos no ponto zero para transmitir zero ou um. O um binário é representado por um pulso de 120Khz no primeiro cruzamento e uma ausência de pulso no segundo, já o zero binário é representado por uma ausência de pulso no primeiro e um pulso de 120Khz no segundo (Figura 14).

Os endereços X-10 são constituídos por duas partes de quatro bits cada: código de casa (*House Code*) e código de unidade (*Unit Code*). Normalmente, dentro de uma única casa é atribuído o mesmo código de casa aos controladores e aos módulos, sendo o código de unidade utilizado para selecionar cada um desses módulos em particular. O código de casa é utilizado para separar módulos que podem estar instalados em casas ou apartamentos vizinhos, mas que dividem o uso da mesma rede elétrica (FERNANDES, 2000).

Após ter configurado o endereçamento de cada dispositivo é possível detalhar a transmissão de mensagens. Cada mensagem é constituída por treze bits separados em quatro campos: código de início (*Start Code*), código de casa (*House Code*), código de unidade (*Unit Code*) ou código de função (*Function Code*) e um campo de bit de função (*Function Bit*), responsável por definir quando esta sendo transmitido um código de unidade ou um código de função.

A mensagem X-10 é sempre transmitida aos pares em um pacote de 26 bits sem intervalo de ciclos de potência entre as mensagens. Desse modo cria-se redundância na transmissão por segurança deixando a critério do módulo receptor validar uma a outra. Já um pacote completo necessita o intervalo de três ciclos de potência (Figura 16) antes que se possa enviar outro pacote. A exceção dessa regra são os comandos BRIGHT e DIM. No caso desses dois comandos não existem intervalos entre os pacotes (TEZA, 2002, p. 74).

#### 2.4.2. BLUETOOTH

A unidade fundamental de um sistema Bluetooth é uma *piconet*, que consiste em um nó mestre e até sete nós escravos ativos localizados dentro de uma distância de 10 metros. É possível utilizar-se de diversas *piconets* dentro da mesma sala e elas podem até mesmo serem conectadas por um nó de ponte. Quando se tem diversas redes *piconets* conectadas umas as outras a estrutura passa a ser chamada de *scatternet* (TENENBAUM, 2003, p.331).

A especificação do *Bluetooth* versão 1.1 identifica treze aplicações específicas e fornece diferentes pilhas de protocolos para cada uma. As treze aplicações, chamadas perfis estão contidas na Tabela 1. O perfil de acesso genérico não é realmente uma aplicação, mas sim a base sobre o qual são implementadas as aplicações reais. Possui como principal função fornecer um meio para estabelecer e manter enlaces seguros entre o mestre e os escravos. Também é relativamente genérico o perfil de descoberta de serviços, que é responsável por descobrir quais são os serviços que outros dispositivos têm a oferecer. É esperado que todos os dispositivos de *Bluetooth* implementem esses dois perfis, os demais são opcionais (TANENBAUM, 2003, p. 332).

**Tabela 1:** Os perfis do Bluetooth

Nome	Descrição
Acesso genérico	Procedimento para gerenciamento de enlaces
Descoberta de serviços	Protocolo para descobrir serviços oferecidos
Porta serial	Substitui um cabo de porta serial
Intercâmbio genérico de objetos	Define o relacionamento cliente/servidor para movimentação de objetos
Acesso de LAN	Protocolo entre um computador móvel e uma LAN fixa
Rede dial-up	Permite que um notebook se conecte através de um telefone móvel
FAX	Permite que um equipamento de fax móvel se comunique com um telefone móvel
Telefonia sem fio	Conecta um aparelho telefônico à sua estação-base local
Intercomunicador	Intercomunicação digital
Fone de ouvido	Permite a comunicação de voz sem o uso das mãos
<i>Push</i> de objetos	Fornecer um meio para intercambiar objetos simples
Transferência de arquivos	Fornecer um recurso mais geral de transferência de arquivos
Sincronização	Permite sincronizar um PDA com outro computador

A estrutura de camadas do *Bluetooth* não segue o modelo OSI, o modelo TCP/IP, o modelo 802 ou qualquer outro modelo conhecido. Entretanto, o IEEE está trabalhando na modificação do *Bluetooth* para adaptá-lo melhor ao modelo 802. A camada inferior é a camada física de rádio. Ela lida com a transmissão e a modulação de rádio e pode até corresponder à camada física nos modelos OSI e 802. A camada de banda-base é de certa forma análoga à subcamada MAC, mas também inclui elementos de camada física. Ela lida com a maneira como o mestre controla os *slots* de tempo e como esses *slots* são agrupados em quadros (TANENBAUM, 2003, p.334).

A pilha de protocolos *Bluetooth* é dividida em duas partes, são elas: *Bluetooth Host* e *Bluetooth Controller*. O HCI (*Host Controller Interface*) é uma interface padronizada que possui como objetivo possibilitar a comunicação entre essas duas partes da divisão da pilha do protocolo. O *Bluetooth Host* é o software que implementa os perfis presente nesse padrão. Ele é executado nos dispositivos, ou seja, está integrado ao software do sistema ou ao sistema operacional. Por sua vez o *Bluetooth Controller* é um módulo de hardware que normalmente corresponde a um adaptador USB (*Universal Serial Bus*) ou um PCMCIA (*Peripheral Component Microchannel Interconnect Architecture*). Sua comunicação com o *host* é realizada através do HCI a não ser se estas duas partes da pilha vierem integradas em um único dispositivo (KUMAR, KLINE & THOMPSON, 2003).

### 3. APLICAÇÃO

Para o desenvolvimento do sistema proposto, o projeto foi dividido em três etapas, são elas: implementação do módulo conversor, implementação do módulo controlador e testes.

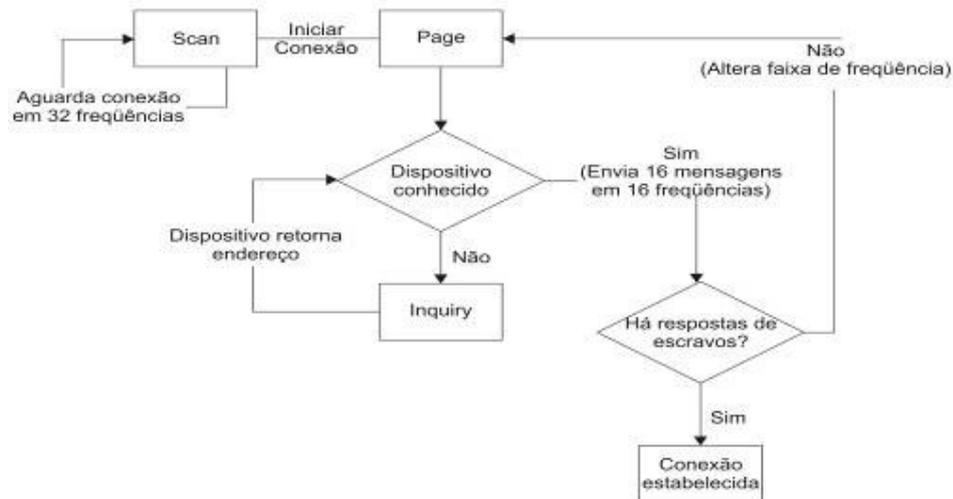
#### 3.1. MÓDULO CONTROLADOR

Esse módulo é constituído do aparelho celular Nokia N95 e do aplicativo desenvolvido com a finalidade de gestão doméstica. Para seu desenvolvimento foi utilizado o software NetBeans IDE 6.9.1 acrescido do Sun Java™ Wireless Toolkit 2.5.2\_01 para CLDC. Sistema operacional Windows 7 fornecido pela UNIVALI através do programa MSDNAA (Microsoft Developer Network Academic Alliance)

O celular Nokia N95 usufrui do sistema operacional Symbian. O Symbian OS (*Operation System*) é um sistema operacional para ambientes móveis e sua linguagem nativa é o C++. Entretanto, ele implementa a API JSR-82. Essa API (*Application Programming Interface*) é somente disponibilizada com dispositivos móveis que fazem parte do MIDP 2.0 e que possuem a tecnologia *Bluetooth*. Essa API se baseia em um *framework* do MIDP chamado de *Generics Connection Framework* para prover a comunicação. A conexão *Bluetooth* é obtida por meio desse *framework*, o qual fica localizado no *javax.microedition.io.\**.

A API JSR-82 é composta de dois pacotes: o *javax.bluetooth.\** e o *javax.obex.\**. Para o desenvolvimento desse módulo somente o primeiro pacote citado foi utilizado. Nesse pacote existem duas classes as quais são essenciais, são elas: *javax.bluetooth.LocalDevice* e *javax.bluetooth.RemoteDevice*. A partir da utilização dessas classes, se torna possível realizar os passos necessários para que aja comunicação através do *Bluetooth*. O funcionamento básico de uma conexão *Bluetooth* consiste em três estados: *scan*, *page* e *inquiry*. Quando o dispositivo não está conectado em uma *piconet*, ele se encontra no estado de *scan*. Nesse estado, o dispositivo procura por mensagens a cada 1,28 segundos em até 32 frequências. Quando um dispositivo deseja estabelecer a conexão com outro, ele envia 16 mensagens *page* idênticas divididas em 16 frequências. Caso não ocorra a resposta dos escravos o mestre retransmite as mensagens *page* para outras 16 frequências.

Se o mestre não conhece o endereço do escravo, antes de enviar uma mensagem *page* ele envia mensagens *inquiry* disponibilizando os serviços para essa conexão. Na Figura 4 é possível verificar os passos necessários para o início de conexão *Bluetooth*. Perante esse projeto o módulo controlador opera em modo escravo, ou seja, o módulo conversor é o mestre da conexão.



**Figura 4:** Processo para um dois dispositivos estabelecerem a conexão *Bluetooth*.

Depois de realizada a conexão ao módulo conversor o módulo controlador apresenta uma interface contendo os cômodos contidos na residência. Determinado o cômodo, o aplicativo expõe os objetos automatizados contido no mesmo.

Na Figura 5 é possível verificar as interfaces desenvolvidas para esse módulo.



**Figura 5.** Interfaces do módulo controlador: (a) Splashscreen, (b)Inicial, (c)Aguardando busca, (d) Busca Completa, (e) Seleção de cômodos, (f) Seleção de dispositivos.

Ao acionar a mudança de estado (ligado/desligado/intensidade) do objeto desejado o módulo controlador encaminha ao módulo conversor um pacote contendo 3 bytes. O primeiro byte corresponde à sinalização do início de transmissão (*Start TX*), o segundo byte ao comando a ser transmitido e o terceiro byte sinaliza o fim da transmissão (*End TX*). Para confirmação do recebimento correto do pacote esse módulo gera uma *Thread* a qual aguarda o *flag* de sinalização do módulo conversor. Caso seja retornado 0 o comando foi correto, caso seja 1 o pacote é retransmitido.

### 3.2. MÓDULO CONVERSOR

Esse módulo é responsável por converter o pacote de dados emitido pelo módulo controlador para o padrão X-10 e encaminhá-lo ao quadro de automação da residência. Para o desenvolvimento do mesmo inicialmente foi realizada um levantamento para determinar qual interface de hardware seria utilizada para gerar o sinal *Bluetooth*. O levantamento dessas interfaces foi realizado sobre os seguintes critérios: disponibilidade

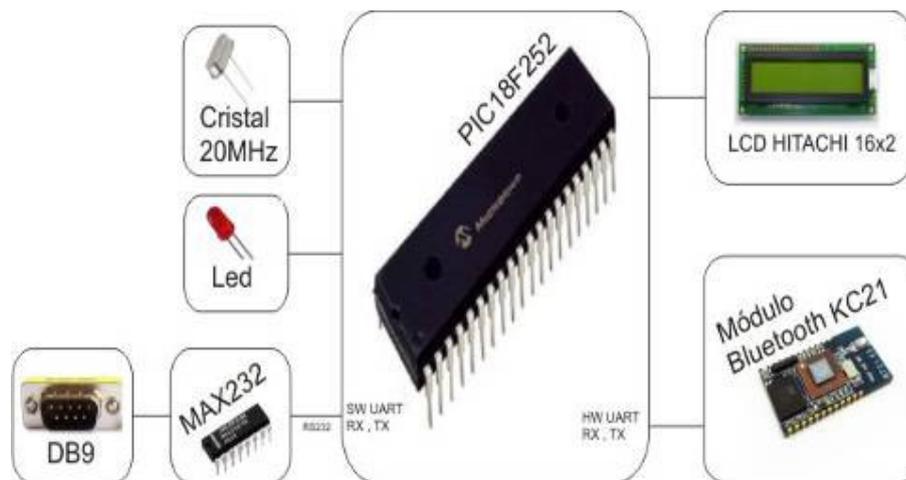
no mercado brasileiro, custo, característica de poder trabalhar como mestre na comunicação e que sejam compatíveis com microcontrolador PIC. Na Tabela 2 é possível verificar um breve resumo dos resultados obtidos.

**Tabela 2:** Comparação de interfaces de hardware para *Bluetooth*

	FCC Class	Modo de Operação	Perfis suportados	Custo (R\$)
GL-6B	Class 2	Master e Slave	Não especificado	100,00
KC-21	Class 2	Máster e Slave	SPP, SDAP, GAP e DUN	130,00
RBT-001	Class 2	Master e Slave	SPP, SDAP e GAP	310,00
RN-41	Class 1	Máster e Slave	SPP, GAP, DUN	210,00
SIMBAD	Class 3	Máster e Slave	SPP	90,00

Depois de ter a relação das interfaces disponíveis no mercado nacional foi escolhido o módulo KC-21 do fabricante Wirefree. Esse módulo possui uma antena interna, que conforme a Tabela 1 pode-se enquadrá-la como classe 2. Desse modo possui limite de alcance de 30 metros e oferece comunicação serial de velocidade máxima de 115 KBaud. Para seu funcionamento é necessário uma tensão de 3,3 volts. Ele possui 14 pinos de entrada e saída de propósito geral e memória flash de 8 Mbits.

A seguir, na Figura 6, em forma de diagrama de bloco são apresentados os componentes de hardware utilizado no módulo conversor.



**Figura 6:** Diagrama de bloco do módulo conversor.

Na Figura 6, ao lado esquerdo do microcontrolador consta o conector DB9 fêmea, o qual é responsável por realizar a conexão física do cabo serial ao módulo, logo após segue o CI MAX232, esse realiza a conversão do nível de tensão RS232 para TTL ou vice e versa. Já o led presente no módulo possui a função de sinalizar quando o módulo *Bluetooth* está apto a receber conexões e o cristal define a frequência de funcionamento do microcontrolador, conseqüentemente os períodos de *clock*. No lado direito do microcontrolador possui um display LCD de 16x2, nele é escrito os dados recebidos do módulo controlador e os dados a serem enviados ao quadro de automação residencial. Por fim, no diagrama consta o módulo *Bluetooth* KC21 da Wirefree definido anteriormente.

Depois de definido o hardware necessário, foi desenvolvido o *firmware* para seu controle. Para o desenvolvimento do *firmware* presente no PIC18F252 foi utilizado o

compilador CCS versão 4.049. Esse *firmware* possui como principal função monitorar os pacotes que chegam através da interface *Bluetooth* e converte-los. Na Figura 7 é apresentado o fluxograma que representa as principais etapas em que os dados são manipulados em tempo de execução do código gerado.

A conversão dos dados procede à seguinte maneira: é recebido um pacote contendo 3 bytes encaminhado pelo módulo controlador. Esses 3 bytes são convertidos a um pacote de 13 bits. Após o pacote é duplicado de modo a formar o comando que será encaminhado ao quadro de automação, esse, segundo o protocolo X-10 deve conter 26 bits para que seja possível acionar os dispositivos e também para realizar a validação do mesmo. (FERNANDES, 2000).

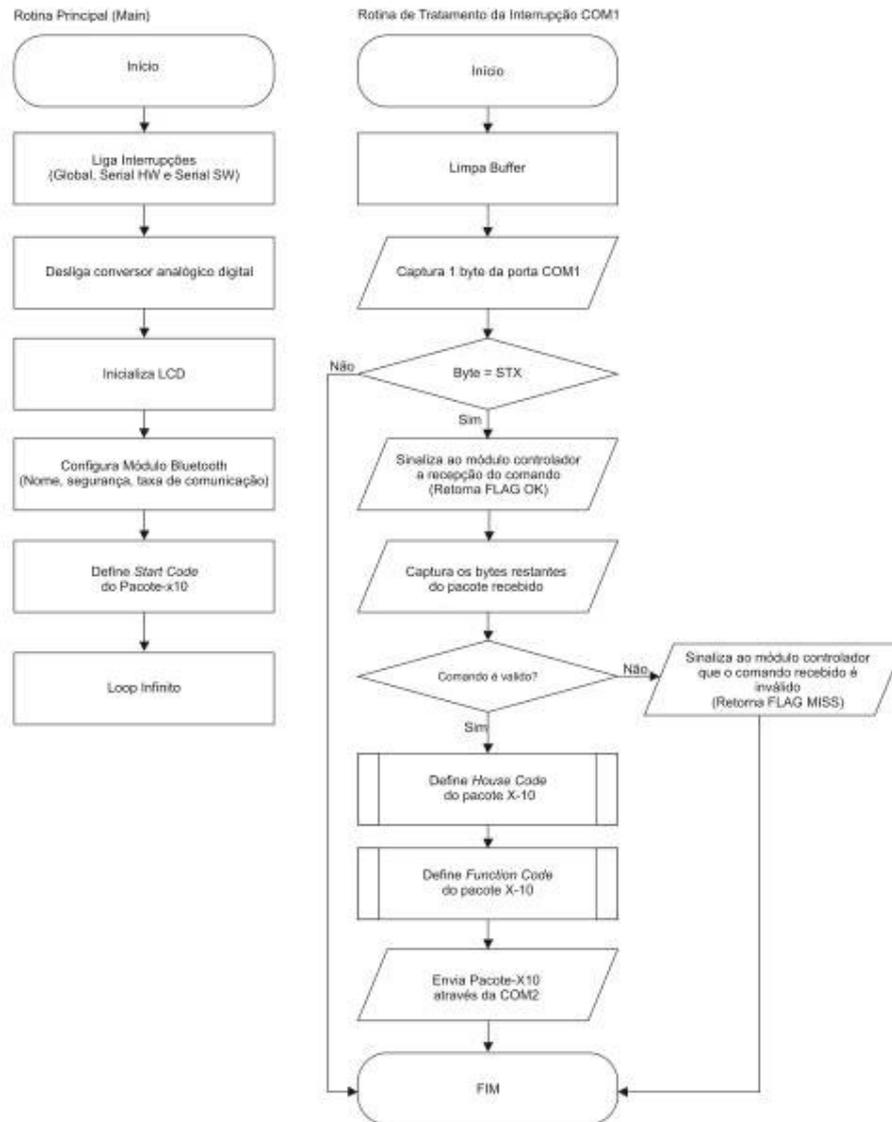


Figura 7: Fluxograma do firmware

### 3.3. TESTES

Para poder validar o desenvolvimento do módulo controlador e do módulo conversor foi desenvolvido um terceiro software, esse nomeado de módulo de teste. O módulo de teste foi desenvolvido na linguagem J2SE, possui como principal

funcionalidade receber dados através da serial no padrão elétrico RS232 e interpretar os comandos no formato do pacote descrito no protocolo X-10. Esse software simula uma residência (Planta Baixa) contendo diversos cômodos os quais possuem objetos habituais presentes no mesmo. A interface desse módulo pode ser visualizada na Figura 8.

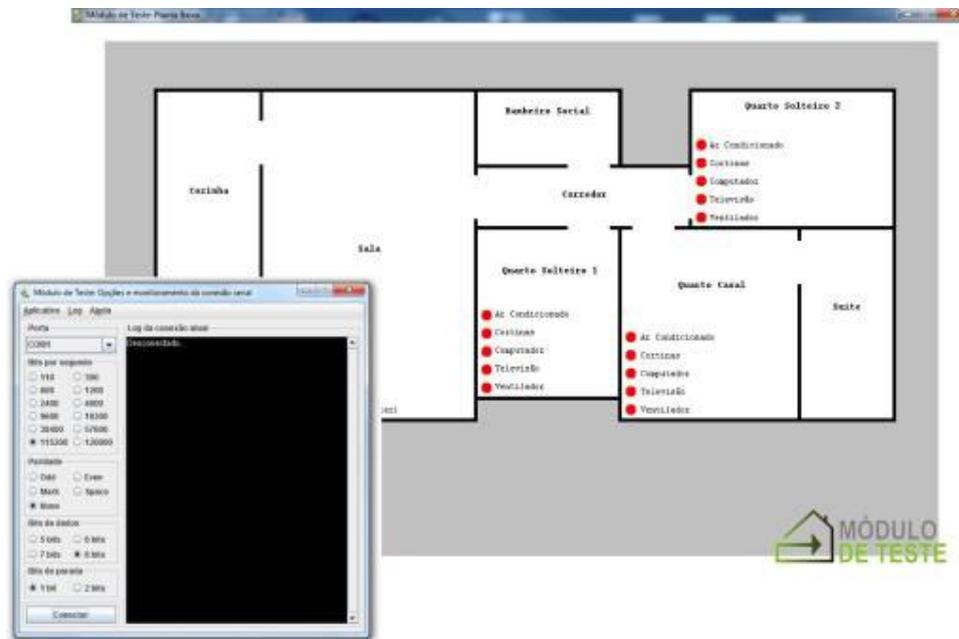


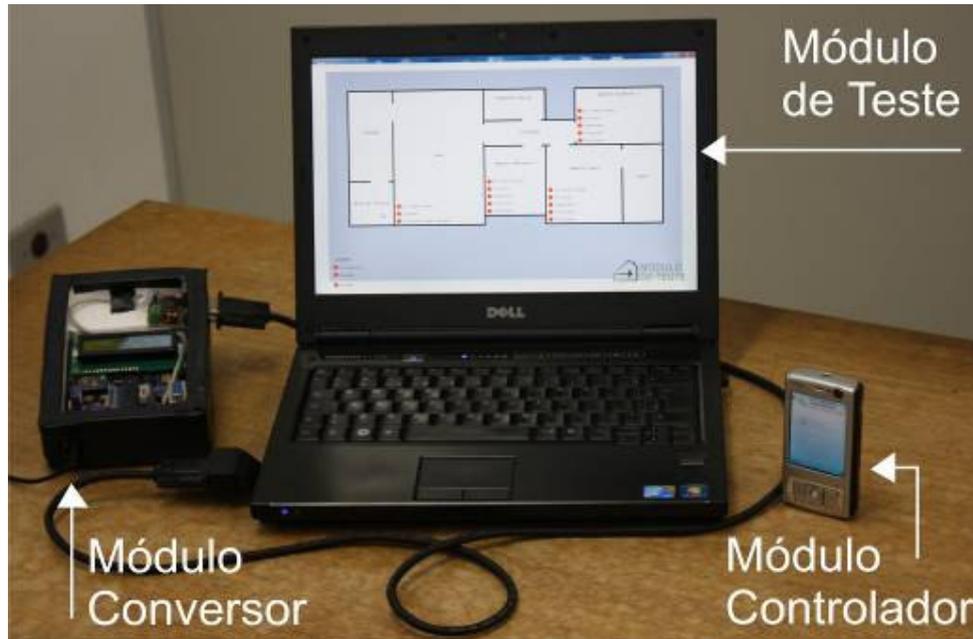
Figura 8: Interface do módulo de teste.

Em sua implementação foi utilizado a API RXTXComm 2.1, essa API é baseada na API Javacomm distribuída pela própria SUN.

Ao iniciar o recebimento, é aguardada a chegada dos 26 bits que compõem o pacote X-10 os quais foram enviados pelo módulo conversor. Após o recebimento esse pacote é dividido em duas partes iguais, desse modo se tornar possível realizar a verificação do comando. Essa verificação é realizada comparando os 13 primeiros bits com os 13 últimos bits. Caso eles sejam iguais é extraído os 4 bits referentes ao *HouseCode* e os 4 bits referentes ao *FunctionCode*, se forem diferentes é encaminhando ao módulo conversor uma solicitação de reenvio do pacote. Depois de extraído esses bits se torna possível realizar a mudança de estado dos objetos contidos na residência. Os objetos contidos nos cômodos são representados através de um círculo vermelho quando se encontram no estado de desligado e um círculo verde quando se encontram no estado de ligado. As luzes dos cômodos são representadas através da cor a qual se encontra o retângulo que defini o próprio cômodo, ou seja, se o cômodo estiver na cor branca à iluminação se encontra desligada, caso esteja na cor amarela à iluminação se encontra no estado de ligada.

Após o desenvolvimento do módulo de teste a validação do projeto como um todo (Figura 9) decorreu da seguinte maneira: no módulo controlador, foi utilizada uma interface de log contida no mesmo, o qual descreve o endereço MAC do dispositivo local, o endereço MAC do dispositivo remoto, a URL de conexão e os comandos que estão sendo enviados. Ao chegar o pacote no módulo conversor, esse dispõe de um botão para depuração que ao ser pressionado apresenta no display LCD o pacote recebido e o pacote a ser transmitido ao módulo de teste. Já no módulo de teste, pode-se

verificar o pacote recebido através da tela de *log* ou do acionamento virtual dos objetos pertencentes à casa. Após realização dos passos descritos acima foi constatado que o sistema funciona do modo esperado.



**Figura 9:** Foto do sistema completo.

Também foram realizados testes de forma a averiguar para que distância máxima a conexão *Bluetooth* permanece ativa. Esse teste consistiu em estabelecer a conexão do módulo controlador ao módulo conversor e depois afastar o módulo controlador até observar a interrupção da conexão. Foi constatado que em ambiente aberto a conexão é perdida aos 22 metros e em ambiente fechado aos 13 metros.

### 3. CONCLUSÕES

Foi apresentada nesse trabalho uma alternativa de controle para uso doméstico o qual é possível realizar adaptações para que se possa utilizar sobre diversos protocolos de comunicação. O aplicativo desenvolvido para o módulo controlador foi testado no celular Nokia N95, Nokia N98, Nokia 6822 e Motorola Defy MB525. O Motorola citado opera com sistema operacional Android e a para sua utilização no mesmo deverá ser reprogramado os métodos de conexão e controle *Bluetooth*, isso é devido ao fato do sistema restringir o acesso direto ao hardware e a implementação da pilha ser realizada através da API *android.bluetooth.\**, essa disponibilizada para J2ME.

Foi constatado que o sinal *Bluetooth* sofre diversas interferências quando localizados próximos a roteadores *Wi-Fi* e telefones sem fio que operam na mesma faixa de frequência. Desse modo há um consumo maior de bateria do dispositivo em função de realizar um número maior de saltos entre as frequências possíveis para a conexão e há um número maior de retransmissão automática dos comandos solicitados, provendo um tempo maior para acionamento dos dispositivos. Entretanto os ensaios práticos foram satisfatórios ratificando o conceito do princípio de funcionamento da proposta de aplicação por radio frequência.

#### 4. REFERÊNCIAS

ALIEVI, C. A. **Automação residencial com utilização de controlador lógico**. 2008. 84 f. Trabalho de Conclusão de Curso (Especialização) – Instituto de Ciências Exatas e Tecnológicas, Centro Universitário FEEVALE, Novo Hamburgo, 2008.

AURESIDE, Associação Brasileira de Automação Residencial. **A casa da Microsoft**. Disponível em <http://www.aureside.org.br/temastec/default.asp?file=concbasicos03.asp>. Acessado em 02/08/2010.

BOLZANI, Caio Augustus Morais, **Residências inteligentes**. São Paulo, Ed. Editora e Livraria da Física, 2007.

DÍAZ, Oscar Moreno; PARDO, Jose Luiz Aller; PULIDO, Ivan Mariano. **Domótica – Intalación y simulación**. Disponível em <http://bibliotecnica.upc.es/bustia/arxiu/40441.pdf>. Acessado em 07/09/2010.

FERNANDES, Pedro Miguel de Miranda. **Aplicações domóticas para cidadãos com paralisia cerebral**. Disponível em: <http://portal.ua.pt/bibliotecad/default1.asp?Obra=28&H1=2&H2=0> Acessado em 04/08/2010.

ROQUE, Eng António. **Introdução a domótica**. Publicado na Revista O Electricista, nº1 Jul, Agos e Set de 2002. Disponível em: <http://www.antonioroque.com/textos.asp?idCat=11&idArtigo=12> Acesso em 08/08/2008.

TANENBAUM, Andrew S. **Redes de Computadores**. Ed. Elsevier. Tradução autorizada do idioma inglês da edição publicada por Pretince Hall. Rio de Janeiro, 2003.

TERUEL, E. C. **Uma proposta de framework para sistemas de automação residencial com interface para WEB**. 2008. 158 f. Dissertação (Mestrado em Tecnologia: Gestão, Desenvolvimento e Formação) – Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2008.