

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

RAFAEL RAUTER HERESCU

**Automação Residencial em Redes sem Fio**

Trabalho de Diplomação.

Prof. Dr. João César Netto  
Orientador

Porto Alegre, dezembro de 2009.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do ECP: Prof. Gilson Inácio Wirth

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## **AGRADECIMENTOS**

Após a conclusão de mais uma etapa importante da minha vida, gostaria, inicialmente, de agradecer a Deus por ter me dado força, coragem e saúde em todos os momentos que precisei.

Agradeço aos meus pais Sergio Gabriel Herescu e Vivian Célia Rauter Herescu por todo o carinho e apoio incondicionais durante toda minha vida e por sempre me ensinarem que não há investimento maior do que a educação.

Agradeço ao meu irmão Eduardo Rauter Herescu pela sua paciência e apoio em todos os momentos.

Meus sinceros agradecimentos a todos da minha família que sempre me deram ânimo e sempre torceram por mim.

Também agradeço aos meus amigos por estarem presentes nos momentos bons e ruins, pois sem eles com certeza tudo seria mais difícil e menos divertido.

Ao meu orientador, João César Netto, agradeço pela paciência e pelos ensinamentos.

Agradeço às pessoas do PRAV pela oportunidade de trabalho na área de pesquisa, que me propiciou maior experiência acadêmica e profissional.

Finalmente, agradeço ao leitor pela oportunidade de compartilhar o meu conhecimento.

# SUMÁRIO

<b>AGRADECIMENTOS</b> .....	<b>3</b>
<b>SUMÁRIO</b> .....	<b>4</b>
<b>LISTA DE ABREVIATURAS E SIGLAS</b> .....	<b>6</b>
<b>LISTA DE FIGURAS</b> .....	<b>8</b>
<b>LISTA DE TABELAS</b> .....	<b>9</b>
<b>RESUMO</b> .....	<b>10</b>
<b>ABSTRACT</b> .....	<b>11</b>
<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 Contextualização .....	12
1.2 Motivação .....	13
1.3 Objetivos .....	14
<b>2 OPÇÕES DE IMPLEMENTAÇÃO</b> .....	<b>15</b>
2.1 Análise de protocolos de redes sem fio.....	15
2.2 WirelessHART e ZigBee .....	17
2.3 Enlace IEEE 802.15.4 e WirelessHART .....	17
<b>3 IEEE 802.15.4</b> .....	<b>19</b>
3.1 Ocupação de banda e modulação.....	19
3.2 Controle de acesso ao meio.....	20
3.3 Consumo de energia .....	20
3.4 Topologias Estrela e Ponto-a-Ponto.....	20
3.5 Tipos de quadros .....	21
3.5.1 Superframe .....	21
3.5.2 Beacon Frame .....	22
3.5.3 Data Frame.....	22
3.5.4 Acknowledgment Frame.....	23
3.5.5 MAC Command Frame .....	23
<b>4 ARQUITETURA DO HARDWARE</b> .....	<b>24</b>
4.1 Placas de rádio do laboratório LAIS .....	25
4.2 Placas de rádio “Design Reference” da empresa Freescale .....	26
4.2.1 Etapas de fabricação .....	26
4.2.2 Custos do projeto.....	28
<b>5 ARQUITETURA DO SOFTWARE</b> .....	<b>29</b>
5.1 Ambiente de desenvolvimento.....	30
5.1.1 CodeWarrior .....	30
5.1.2 API IEEE 802.15.4.....	31
5.1.2.1 Dispositivo de funções reduzidas .....	32
5.1.2.2 Dispositivo de funções completas .....	32
5.1.2.3 Interface MLME.....	32
5.1.2.4 Interface MCPS .....	33

5.1.2.5	Interface ASP .....	33
5.1.3	Porte da API.....	33
<b>5.2</b>	<b>Software dos módulos.....</b>	<b>33</b>
5.2.1	Coordenador PAN.....	35
5.2.1.1	Inicialização do coordenador .....	35
5.2.1.2	Mensagem proveniente da interface MLME .....	37
5.2.1.3	Mensagem proveniente da interface MCPS.....	37
5.2.1.4	Mensagem proveniente da porta RS-232.....	37
5.2.2	Nodos genéricos .....	38
5.2.2.1	Inicialização do nodo genérico.....	39
5.2.2.2	Mensagem proveniente da interface MLME .....	39
5.2.2.3	Mensagem proveniente da interface MCPS.....	39
5.2.2.4	Tempo de “polling” .....	40
<b>5.3</b>	<b>Software de interface, e comunicação com o coordenador PAN .....</b>	<b>40</b>
5.3.1	Software de comunicação entre a interface e o coordenador PAN.....	41
5.3.2	Software de interface “web” .....	41
<b>5.4</b>	<b>Regra de construção das mensagens.....</b>	<b>42</b>
<b>6</b>	<b>TESTE E VALIDAÇÃO .....</b>	<b>43</b>
<b>6.1</b>	<b>Uso da biblioteca SMAC .....</b>	<b>44</b>
<b>6.2</b>	<b>Resultados .....</b>	<b>44</b>
6.2.1	Tempo de transmissão de dados de um nodo ao outro .....	45
6.2.2	Tamanho de código .....	46
6.2.3	Analisador espectral .....	46
<b>7</b>	<b>CONCLUSÃO .....</b>	<b>48</b>
	<b>REFERÊNCIAS.....</b>	<b>50</b>
	<b>ANEXO A – ESQUEMÁTICOS DOS MÓDULOS “DESIGN REFERENCE” ....</b>	<b>52</b>
	<b>ANEXO B – ESQUEMÁTICO DO MÓDULO DO LABORATÓRIO LAIS .....</b>	<b>55</b>

## **LISTA DE ABREVIATURAS E SIGLAS**

ANSI	American National Standards Institute
API	Application Programming Interface
ASK	Amplitude Shift Keying
ASP	Application Support Package
BDM	Background Debug Mode
BPSK	Binary Phase Shift Keying
CAP	Contention Access Period
CFP	Contention Free Period
CGI	Common Gateway Interface
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DSSS	Direct Sequence Spread Spectrum
GPIO	General Purpose Input/Output
GTS	Guaranteed Time Slot
HART	Highway Addressable Remote Transducer
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial, Scientific and Medical
LED	Light Emitting Diode
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Media Access Controller
MCPS	MAC Common Part Sublayer- Service Access Point
MCU	Micro Control Unit
MLME	MAC Sublayer Management Entity
MR	Modelo de Referência
O-QPSK	Offset – Quadrature Phase Shift Keying

OSI	Open Systems Interconnection
PAN	Personal Area Network
PHY	Physical Interface Transceiver
SMAC	Simple Media Access Controller
SPI	Serial Peripheral Interface
TDMA	Time Division Multiple Access
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

## LISTA DE FIGURAS

<i>Figura 2.1: Camadas de níveis inferiores do MR-OSI.</i>	15
<i>Figura 2.2: Protocolo HART no modelo de referência OSI ([3]).</i>	18
<i>Figura 3.1: Sinal modulado com DSSS ([8]).</i>	20
<i>Figura 3.2: Topologia Estrela.</i>	21
<i>Figura 3.3: Topologia ponto-a-ponto.</i>	21
<i>Figura 3.4: Superframe ([2]).</i>	22
<i>Figura 3.5: Beacon frame ([2]).</i>	22
<i>Figura 3.6: Data frame ([2]).</i>	22
<i>Figura 3.7: Acknowledgment frame ([2]).</i>	23
<i>Figura 3.8: MAC command frame ([2]).</i>	23
<i>Figura 4.1: Diagrama em blocos em nível de sistema ([12]).</i>	24
<i>Figura 4.2: Módulo de arquitetura complexa.</i>	25
<i>Figura 4.3: Módulo de arquitetura simplificada.</i>	25
<i>Figura 4.4: Módulo “design reference”.</i>	27
<i>Figura 5.1: Estrutura do sistema.</i>	29
<i>Figura 5.2: Diagrama em blocos da comunicação dos softwares.</i>	30
<i>Figura 5.3: CodeWarrior IDE.</i>	31
<i>Figura 5.4: Circuito OpenSource BDM.</i>	31
<i>Figura 5.5: Diagrama em blocos da API IEEE 802.15.4 ([14]).</i>	32
<i>Figura 5.6: Mensagens de associação.</i>	34
<i>Figura 5.7: Fluxograma do laço principal do coordenador PAN.</i>	36
<i>Figura 5.8: Fluxograma do laço principal do nodo genérico.</i>	38
<i>Figura 5.9: Diagrama em blocos do software intermediário.</i>	41
<i>Figura 5.10: Página “web”.</i>	42
<i>Figura 6.1: Analisador de espectro.</i>	47



## LISTA DE TABELAS

<i>Tabela 2.1: Classificação dos protocolos de comunicação sem fio.</i> .....	15
<i>Tabela 2.2: Comparativo entre os diferentes protocolos de redes sem fio.</i> .....	16
<i>Tabela 3.1: Bandas de frequência do padrão IEEE 802.15.4.</i> .....	19
<i>Tabela 3.2: Comandos MAC.</i> .....	23
<i>Tabela 4.1: Custos do projeto.</i> .....	28
<i>Tabela 6.1: Tempos de transmissão de mensagens.</i> .....	45
<i>Tabela 6.2: Tamanho de código dos softwares dos módulos.</i> .....	46

## **RESUMO**

Este projeto visa desenvolver a base do sistema de comunicação entre dispositivos micro-controlados de baixa potência para que possa se estabelecer uma rede entre eles e que através dessa rede, um pacote possa ser roteado entre os dispositivos a fim de que possam alcançar outros dispositivos fisicamente fora de alcance. No caso, a rede será aplicada a um ambiente residencial no qual haverá um dispositivo mestre que terá por objetivo a manutenção e controle da rede e dos demais dispositivos escravos. O propósito é o de fazer os dispositivos interagirem com o ambiente ligando luzes, travando portas, molhando a grama, entre outras funcionalidades.

**Palavras-Chave:** automação residencial, redes sem fio, redes de sensores, IEEE 802.15.4.

# **Home Automation on Wireless Networks**

## **ABSTRACT**

This project intends to develop the basis of a system for the communication between low-powered micro-controlled devices in order to establish a network between them and through that network to route a packet between the devices so it can reach other devices that are physically unreachable. In this project, the network will be applied to a residential environment in which there will be a master device that has the purpose of maintaining and controlling the network and the other slave devices. This way the devices can interact with the environment turning lights on, locking doors, wetting the grass, and many other features.

**Keywords:** home automation, wireless networks, sensor networks, IEEE 802.15.4.

# 1 INTRODUÇÃO

A automação residencial (ou domótica) é um sub-campo de automação predial que define os requisitos específicos de automação de residências, para prover conforto e segurança para os moradores. Ela inclui controle de iluminação, de portas e janelas, de irrigação de jardins, climatização, alimentação de animais, e outras funcionalidades com uma interface amigável ao usuário. Além disso, atualmente, está sendo altamente voltada para eficiência energética.

A competitividade do mercado de trabalho levou as pessoas a passarem cada vez menos tempo em casa, e mais tempo em escritórios. Por este motivo, a exigência de conforto e segurança para se passar tempo com qualidade em casa vem tornando-se cada vez maior.

Apesar da grande evolução das tecnologias de redes permitindo tráfegos de grandes massas de dados, há aplicações que não demandam toda essa banda, como as WSN. Essas redes consistem na conexão entre dispositivos autônomos que utilizam sensores para monitoramento do ambiente e prevêm que os dispositivos estarão em locais próximos uns dos outros. Elas são particularmente importantes na área de automação residencial, por se prestarem muito bem a suprir várias das funcionalidades que a rede demanda.

Os equipamentos utilizados atualmente para tornar residências inteligentes ainda são, em sua maioria, baseados em redes cabeadas. Nestas redes, a interconexão de dispositivos deve ser pensada, preferencialmente, no projeto do prédio, para prever por onde deve ser passado o cabeamento, que começa a se tornar volumoso dependendo do grau de integração que se deseja ter. Mesmo assim, continua sendo um grande problema caso se deseje ter uma futura expansão da rede.

Frente a todos os problemas agregados ao uso de redes cabeadas, as redes sem fio, nas quais o transporte de informação é feito pelo ar, apresenta muitas vantagens. Além disso, os padrões de redes sem fio têm crescido e evoluído de forma rápida nos últimos anos, o que dá margem para análise, comparação e escolha do melhor meio a ser utilizado.

## 1.1 Contextualização

Com a inclusão digital, cada vez mais as pessoas têm a necessidade de estarem conectadas o tempo todo para interagir entre elas, através de telefonia celular, ou para acesso à informação, através da internet. Com os avanços da tecnologia, cada vez mais, o trabalho físico é substituído pelo automatizado. Podemos imaginar que as residências também começam a fazer parte desta grande rede. Casas começam a ser totalmente automatizadas, podendo ser acessadas e controladas pela internet e até mesmo pelo telefone celular.

Nos Estados Unidos e países da Europa, a automação doméstica já está presente em muitas casas. No Brasil, a automação em prédios ainda é mais utilizada do que a de automação residencial. Isso se deve em grande parte a esta tecnologia estar disponível apenas ao público de classe alta e grandes corporações, devido ao seu custo.

Grande parte dos produtos desta área que estão disponíveis no mercado atualmente são importados. As empresas do ramo se limitam a fazer apenas projetos e instalações com o que é oferecido pelo mercado e não com o desenvolvimento de novos equipamentos.

As tecnologias sem fio empregadas na maioria das vezes utilizam os padrões X10, mais usual, e o IEEE 802.11, utilizado em algumas aplicações que exigem maior banda.

O protocolo X10 foi desenvolvido na década de 70. Foi a primeira rede de automação residencial de propósitos gerais desenvolvida e até hoje é largamente utilizada. Apesar deste padrão apresentar uma vantagem crucial que é utilizar o próprio cabeamento de força da casa para transmissão de dados (eliminando assim a necessidade de instalar novos cabos), ele apresenta problemas intrínsecos a esta vantagem como: problema de divisão das redes elétricas, atenuação do sinal ao passar pela linha, introdução de ruído podendo provocar perda de comunicação, problema com dispositivos de baixa potência e outros.

O padrão IEEE 802.11 é muito utilizado em soluções que necessitem de maior banda, como por exemplo, monitoramento através de áudio e vídeo. O problema de utilizar este padrão para as soluções de propósitos gerais, como controles de dispositivos simples, é que há muito desperdício de banda. A potência consumida é muito maior e o custo agregado também. Mas, ainda sim, o maior problema continua sendo que ele não se presta para dispositivos de baixa potência.

## **1.2 Motivação**

A motivação para este projeto vem basicamente de dois fatores: o primeiro é a satisfação pessoal, pois desde a infância sempre pensei em morar em uma casa futurista, em que eu pudesse controlar praticamente tudo. O segundo é da visão de uma possível inclusão de um produto bom no mercado de eletrônicos e, talvez, até mesmo vir a produzir tais dispositivos.

A recente vontade de voltar minha formação para a área de sistemas embarcados também motiva a conclusão deste trabalho, que estará abrangendo toda a área de baixo nível no limiar entre hardware e software.

O desafio e o aprendizado são dois fatores muito importantes. A conclusão de um projeto destes no prazo de dois semestres leva a uma satisfação muito grande como engenheiro e como pessoa. O aprendizado irá servir para muitas áreas do curso como: redes, sistemas embarcados, sistemas de tempo-real, etc., ao mesmo tempo em que poderá prover um grande diferencial como profissional.

### 1.3 Objetivos

O objetivo deste projeto é construir a base do sistema de comunicação entre dispositivos micro-controlados de baixo custo (que serão chamados de nodos) que possam ser facilmente reprogramados, de acordo com a necessidade de cada aparelho elétrico por eles controlado. Com isto, para cada diferente aparelho a alteração necessária no nodo será mínima e este será de fácil instalação.

Existem três diferentes tipos de arquiteturas da rede de comunicação para automação residencial:

- Centralizada: há uma central de processamento que recebe informações de sensores, as processa e comanda os atuadores.
- Distribuída: todo o processamento do sistema é distribuído por todos os módulos que consistem em sensores e atuadores.
- Mista: combinação das outras duas.

A escolhida para este projeto será a mista. A idéia inicial é que o sistema tenha uma central de processamento que servirá para três fins: autenticação dos nodos na rede, controle dos mesmos e interface com o usuário. Cada nodo terá seu conjunto de sensor e atuador, e deverá ser capaz de controlar sozinho o aparelho a que ele está ligado. A central será um computador pessoal, com uma placa de rádio ligada a ele.

Talvez um sistema centralizado servisse a este propósito. Mas, outro objetivo deste projeto é que a rede possa assumir uma largura, teoricamente, infinita. Cada nodo deve ter inteligência suficiente para conseguir retransmitir pacotes de outros nodos até a central, da central até outros nodos, e também pacotes entre os nodos. Além disso, dependendo do dispositivo que o nodo está controlando, ele terá que fazer algum processamento adicional.

Isso se deve ao fato de que as redes sem fio de baixa potência possuem diversos problemas de alcance devido a interferências eletro-magnéticas de outros equipamentos e até mesmo paredes, que agem como atenuadores no sinal. Dando aos nodos a capacidade de retransmitirem os pacotes, com o conhecimento prévio do caminho de menor latência até a central, o problema é minimizado.

A interface com o usuário se dará a partir de uma página “web”, que estará funcionando através de um servidor de HTTP que estará sendo executado na central. Através dessa página, será possível autenticar os nodos e executar a funcionalidade mais usual de controlá-los. Isso também poderá ser feito através da internet.

## 2 OPÇÕES DE IMPLEMENTAÇÃO

Por ser um projeto que inclui características específicas de baixa potência, baixo consumo de banda e uma largura de rede que teoricamente pode se estender ao infinito, é crucial que se estabeleça uma análise tecnológica a fim de determinar qual a melhor e mais adequada tecnologia a ser empregada na implementação.

Para melhor compreender a análise comparativa, iremos nos basear no modelo de referência OSI. Isto é pertinente, considerando que estamos desenvolvendo um projeto baseado no uso de redes e o modelo permite uma visualização abstrata e separada em nível de camadas das diferentes partes que o projeto vai abranger. A Figura 2.1, a seguir, mostra a organização da parte de baixo nível da rede que será abrangida neste documento.

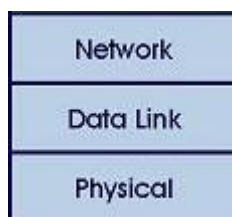


Figura 2.1: Camadas de níveis inferiores do MR-OSI.

### 2.1 Análise de protocolos de redes sem fio

Os objetivos e propósitos dos protocolos e tecnologias de redes sem fio variam muito. Em função disso, a comparação estabelecida a seguir na Tabela 2.1 permite que de certa forma se faça uma escolha simples e direta, não abrindo muito espaço para dúvidas com relação à tecnologia que deve ser adotada neste projeto.

Tabela 2.1: Classificação dos protocolos de comunicação sem fio.

Descrição do Protocolo	Finalidade/Utilização
WirelessHART e ZigBee Padrão IEEE: 802.15.4 Wireless PANs Baixo alcance, na ordem de poucos metros	Tecnologia voltada para a indústria, a ser utilizada em ambientes contendo grande quantidade de clientes em redes de fácil criação / manutenção, altamente dinâmicas, com pouca preocupação com velocidade ou largura de banda e preocupação com o baixo custo e o baixo consumo de energia da comunicação entre dispositivos.  Um cenário típico é o uso em sensores e atuadores.
Bluetooth	Tecnologia criada para comunicação entre telefones celulares e acessórios utilizando sinais de rádio de baixo

Padrão IEEE: 802.15.1 Wireless PAN Alcance médio, na ordem de dezenas de metros	custo, ao invés dos tradicionais cabos. Utilizada em soluções em que se precisa de um consumo de banda considerável e baixa potência.
WiFi Padrão IEEE: 802.11 Wireless LAN Alcance médio, na ordem de dezenas de metros	Tecnologia criada visando a substituição dos cabos em redes domésticas e corporativas, com ênfase na velocidade e uma preocupação menor com custos e consumo de energia – embora seja amplamente utilizada em notebooks. Também é usada em AGVs e robôs industriais. As redes WiFi são comumente utilizadas para a criação de LANs sem a necessidade de conexões físicas.
WiMAX Padrão IEEE: 802.16 Wireless WAN Alcance alto, na ordem de dezenas de quilômetros	<p>Tecnologia que preza pela alta velocidade e pelo alto alcance da rede, o que acarreta custos maiores para a criação e manutenção das redes. Redes WiMAX se propõem a servir um meio de comunicação permanente sobre uma grande área como, por exemplo, uma cidade.</p> <p>Com características de velocidade e alcance semelhantes às encontradas em redes utilizadas por telefones celulares, redes WiMAX são capazes de propiciar um ambiente de conexão constante para um cliente, esteja ele em casa, dentro do carro em movimento, ou no trabalho.</p>

Fonte: [1]. p. 11.

Para este projeto existem dois padrões que se destacam em relação aos demais por possuírem características interessantes, que são os padrões: IEEE 802.15.4 e IEEE 802.15.1. A seguir é apresentada a Tabela 2.2, que tem como objetivo mostrar informações mais específicas sobre cada um desses padrões a fim de se determinar qual será o mais adequado para o uso neste trabalho.

Tabela 2.2: Comparativo entre os diferentes protocolos de redes sem fio.

	WiMAX	WiFi	Bluetooth	WirelessHART /ZigBee
<b>Padrão</b>	802.16	802.11b	802.15.1	802.15.4
<b>Frequência</b>	2 – 66 GHz	5 GHz	2.4 GHz	2.4 GHz
<b>Velocidade</b>	80 Mbps	11 Mbps	1 Mbps	250 Kbps
<b>Alcance</b>	50 Km	100 m	100 m	10 m
<b>Vantagens</b>	Velocidade, Alcance	Custo	Baixo consumo de energia, Custo	Menor consumo de energia, Custo
<b>Desvantagens</b>	Custo	Velocidade	Velocidade, Alcance	Velocidade, Alcance

Fonte: [1]. p. 11.



Considerando o fato de que o fluxo de dados entre os nodos é baixo em função do tráfego de dados entre eles ser basicamente informações de controle, não há muito consumo de banda. Ainda, muitos dos dispositivos que serão controlados não terão necessariamente uma conexão direta à rede de energia elétrica da residência, e a partir disto, a potência dissipada pelos nodos começa a se tornar um fator preocupante. Este pode ser, talvez, o fator que mais deva ser considerado na escolha do protocolo.

O alcance da tecnologia utilizada não é um fator limitante, pois pela característica da própria estrutura da rede é esperado que os nodos estejam em proximidade uns dos outros, permitindo assim que sempre seja possível o estabelecimento de uma rota até a central.

A partir dessa análise é fácil diferenciar o padrão que se enquadra em todos os quesitos do sistema. O mesmo será construído utilizando o padrão IEEE 802.15.4.

## **2.2 WirelessHART e ZigBee**

Atualmente a grande maioria das aplicações sem fio que utilizam tecnologia baseada no padrão IEEE 802.15.4 são construídas utilizando ZigBee. Em setembro de 2007 a HCF (HART Communication Foundation) lançou a especificação HART 7, a qual inclui o WirelessHART, que é o primeiro padrão de comunicação sem fio aberto especificamente voltado para comunicação simples, confiável e segura para plantas e ambientes industriais. Os primeiros produtos confeccionados utilizando esta tecnologia começaram a ser comercializados apenas em torno do segundo quadrimestre de 2008.

Como o WirelessHART foi definido para um ambiente hostil à comunicação de dados no qual os quesitos de segurança, privacidade e confiabilidade são maiores, ele oferece um nível superior de qualidade nestes mesmos quesitos em relação ao ZigBee.

Por ser uma tecnologia muito recente, pouco difundida e com grande potencial de expansão no mercado, o interesse em dominá-la cresce. Em função disso, a opção de implementação deste projeto seguirá a linha do WirelessHART, com simplificações. O trabalho para uso deste protocolo com todas suas camadas por si só já seria alvo para diferentes trabalhos de graduação.

## **2.3 Enlace IEEE 802.15.4 e WirelessHART**

O WirelessHART define um mecanismo de acesso ao canal diferente do definido pelo padrão IEEE 802.15.4. Enquanto que o padrão utiliza o mecanismo CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) (visto mais adiante neste documento), na definição do WirelessHART foram especificados dois diferentes mecanismos: o primeiro baseado em TDMA (Time Division Multiple Access) e o segundo baseado em “token-passing”.

Apesar de o TDMA ser uma das chaves para se conseguir as melhorias do protocolo WirelessHART em relação ao ZigBee mencionadas acima, sua implementação por si só poderia ser objeto alvo de um outro projeto. A adoção desta linha poderia levar a uma mudança total de foco do projeto ou mesmo à sua não conclusão, conforme está sendo aqui descrito.

O objetivo neste escopo é substituir o mecanismo de acesso ao canal do WirelessHART pelo do 802.15.4, porém utilizar a abstração de orientação a comandos definidos nos protocolos HART e WirelessHART e implementar todo o algoritmo de roteamento necessário ao funcionamento da rede.

Com isso, define-se que inicialmente a única camada a ser utilizada do protocolo WirelessHART no modelo de referência OSI, representado pela Figura 2.2, é a que corresponde à camada de aplicação.

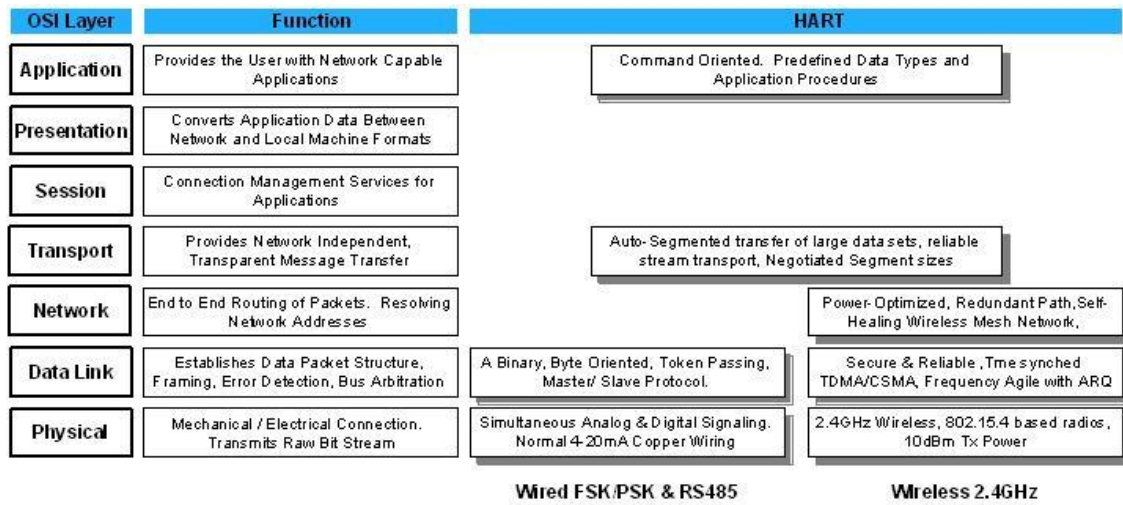


Figura 2.2: Protocolo HART no modelo de referência OSI ([3]).

### 3 IEEE 802.15.4

O padrão IEEE 802.15.4 foi criado em 2003 com o objetivo de definir um padrão de transmissão de dados para WPANs que tinham como quesitos: baixos custo e desempenho. Desde então, se popularizou e atualmente é largamente utilizado por uma grande gama de aplicações, como:

- Automação residencial;
- Controle e monitoramento industrial;
- Monitoramento ambiental e de saúde;
- Entretenimento e brinquedos;
- Emergência e resposta a desastres.

#### 3.1 Ocupação de banda e modulação

O padrão define o comportamento de dois PHYs que operam nas bandas de frequência ISM. Tais frequências estão espalhadas por 27 diferentes canais, que estão divididos em três principais bandas, conforme a Tabela 3.1 abaixo:

Tabela 3.1: Bandas de frequência do padrão IEEE 802.15.4.

Banda	Modulação	Canais	Taxa de dados	Onde
868 – 868,6 MHz	BPSK	1	20 Kbps	Europa
868 – 868,6 MHz	O-QPSK	1	100 Kbps	Europa
868 – 868,6 MHz	ASK	1	250 Kbps	Europa
902 – 928 MHz	BPSK	10	40 Kbps	Estados Unidos
902 – 928 MHz	ASK/O-QPSK	10	250 Kbps	Estados Unidos
2,40 – 2,48 GHz	O-QPSK	16	250 Kbps	Internacional

Fonte: [2]. p. 28

Nota: nas bandas de frequência em que se obtém as taxas mais altas e utiliza-se modulação O-QPSK, também utiliza-se a técnica de espalhamento espectral DSSS (ou sequência direta de espalhamento do espectro).

A utilização da técnica de espalhamento espectral para modular a informação acarreta em uma banda maior necessária para transmitir toda a informação, porém utiliza uma menor densidade de energia espectral para cada sinal. Com isso há menor

interferência na banda de frequência e conseqüentemente há um menor ruído. A Figura 3.1 representa a modulação de um sinal com a técnica DSSS.

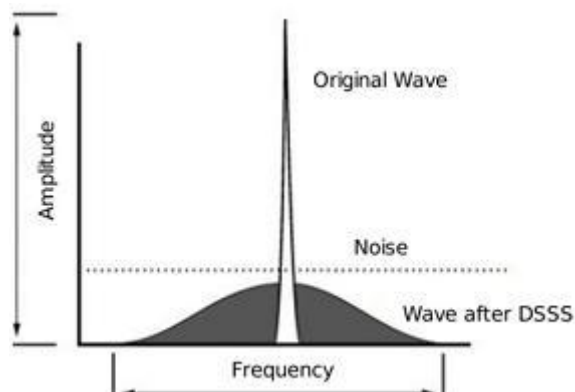


Figura 3.1: Sinal modulado com DSSS ([8]).

### 3.2 Controle de acesso ao meio

Para controle de acesso ao canal, o padrão IEEE 802.15.4 utiliza o mecanismo de CSMA/CA. Cada nodo que desejar transmitir deve primeiramente escutar o meio para ver se há algum outro nodo transmitindo. Se o meio estiver livre, ele pode começar a transmissão, ainda que eventualmente ocorram interferências entre eles no caso de dois começarem a transmitir ao mesmo tempo.

O CSMA/CA ainda ajuda na economia de energia no sentido em que um nodo avisa aos nodos ao seu redor por quanto tempo irá realizar uma transmissão. Com isso evita-se que um nodo tente efetuar múltiplas retransmissões no caso de colisões.

O mecanismo de acesso ao canal é utilizado em combinação com uma das funcionalidades implementadas no padrão em seu nível físico, que é o PLME-ED (Physical Layer Management Entity – Energy Detection). Esta funcionalidade permite que seja sentido o nível de energia em um determinado canal a fim de que se possa começar uma transmissão se o nível de energia no canal for menor ou igual a um limiar pré-definido.

### 3.3 Consumo de energia

O consumo do protocolo é baixo em função de ter poucos ciclos de serviço, ou seja, o transceptor fica num estado de “Sleep” na maior parte do tempo (em média 99% do tempo). Enquanto em funcionamento, o percentual de energia do dispositivo que é destinado ao transceptor depende do tipo de modelo de comunicação que está sendo utilizado.

Se o modo “beacon” estiver sendo utilizado (o qual é o mais comum e utilizado para criação de redes tipo estrela), o tempo gasto para transmitir e receber os quadros “beacon” deve ser somado ao tempo total em que o transceptor encontra-se ligado e conseqüentemente a energia consumida será maior.

### 3.4 Topologias Estrela e Ponto-a-Ponto

Tipicamente o padrão IEEE 802.15.4 permite a construção de redes com dois diferentes tipos de topologias: Estrela (Star) e Ponto-a-Ponto (Peer-to-Peer).

Na topologia estrela há um coordenador PAN, ou mestre de rede, que nesta topologia, atua como um roteador e acaba por ser único. Ele deve estar diretamente conectado a todos os demais nodos da rede e é a única via que eles têm para trocar mensagens entre si. A Figura 3.2 representa esta topologia.

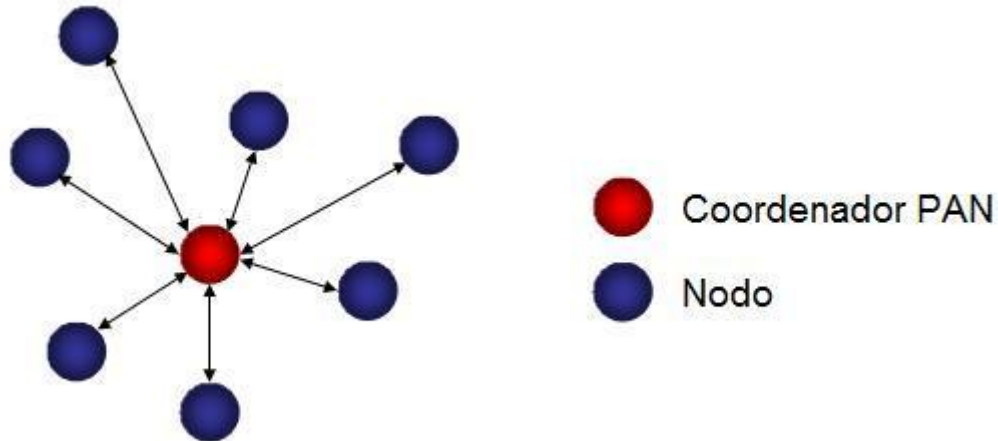


Figura 3.2: Topologia Estrela.

A topologia ponto-a-ponto é a que melhor se enquadra neste projeto, pois diferentemente da topologia estrela, os dispositivos de rede podem se comunicar diretamente, não necessitando de um coordenador PAN para tal. Isso permite que se atinja uma largura de rede muito maior. Neste caso, todos os nodos intermediários devem atuar também como roteadores. Este topologia está representada pela Figura 3.3 abaixo.

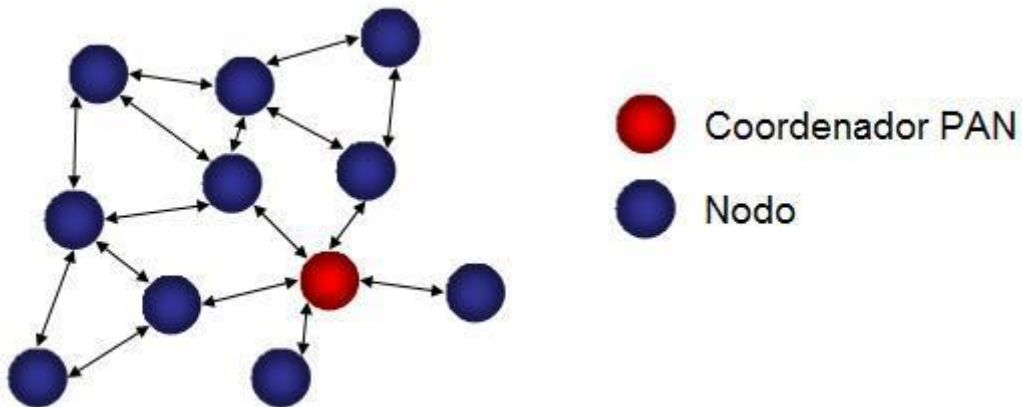


Figura 3.3: Topologia ponto-a-ponto.

### 3.5 Tipos de quadros

O padrão IEEE 802.15.4 define cinco diferentes estruturas de quadro. São elas:

#### 3.5.1 Superframe

Opcionalmente pode ser utilizado pelo coordenador PAN para prover sincronização e informações de configuração para os demais nodos da rede. O superframe é limitado por “beacons” e é dividido em 16 pedaços de tamanhos iguais. Os “beacons” têm por finalidade sincronizar os nodos, identificar o coordenador PAN e descrever a estrutura dos superframes.

O superframe pode conter uma parte ativa e outra inativa. Durante a parte inativa, o coordenador não deve interagir com a rede e pode entrar em modo de baixo-consumo. A parte ativa consiste em um CAP (Contention Access Period) e de um CFP (Contention Free Period). Qualquer dispositivo que quiser se comunicar durante o CAP deve competir com os demais dispositivos utilizando o mecanismo CSMA/CA. Por outro lado, o CFP contém GTSs (Guaranteed Time Slots). Os GTS sempre vêm ao final do superframe, começando no limite do slot imediatamente após o CAP. O formato do superframe pode ser visualizado através da Figura 3.4.

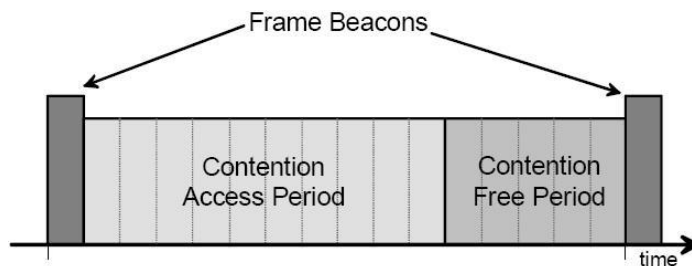


Figura 3.4: Superframe ([2]).

### 3.5.2 Beacon Frame

Utilizado exclusivamente pelo coordenador PAN para transmissão de “beacons”. Serve para sincronização de quadros, controle do ingresso de novos nodos na rede e demarcar os limites do superframe, além de prover um referencial de tempo entre os nodos e o coordenador. A Figura 3.5 mostra o formato deste quadro.

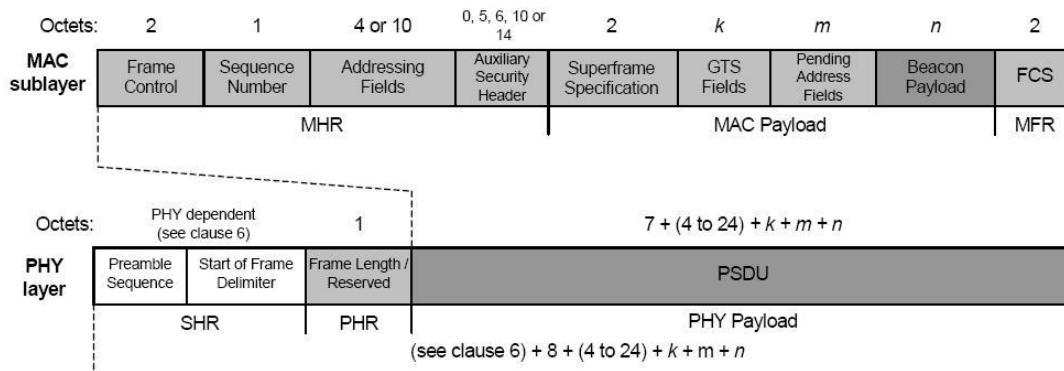


Figura 3.5: Beacon frame ([2]).

### 3.5.3 Data Frame

Utilizado para todas as transferências de dados. Gerado a partir da camada MAC com o objetivo de encapsular os dados oriundos de camadas superiores. O formato deste quadro é representado pela Figura 3.6.

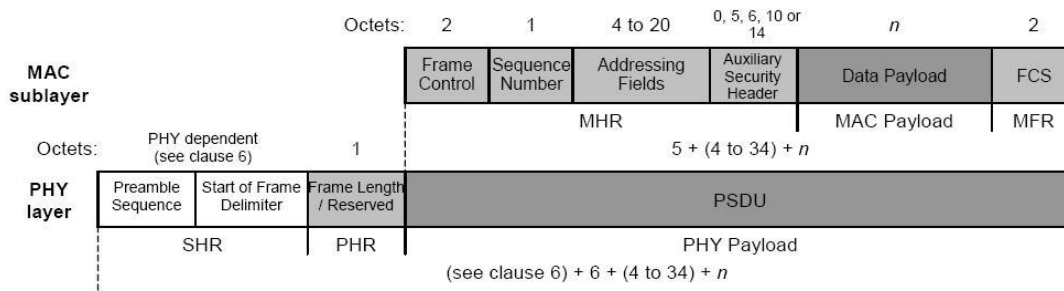


Figura 3.6: Data frame ([2]).

### 3.5.4 Acknowledgment Frame

Utilizado para confirmar ao transmissor o êxito na recepção de um quadro. Este quadro só é gerado caso o CRC coincida com o CRC calculado através dos dados recebidos. Seu formato é representado através da Figura 3.7.

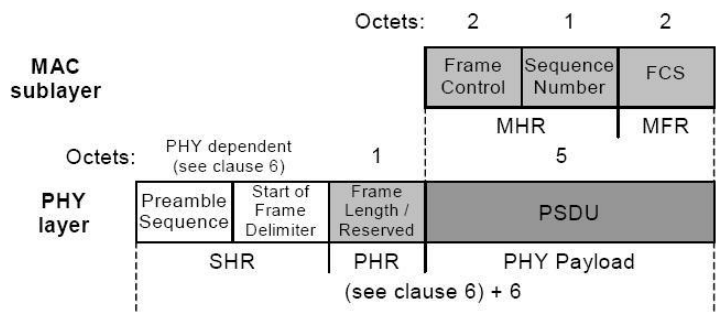


Figura 3.7: Acknowledgment frame ([2]).

### 3.5.5 MAC Command Frame

Utilizado pelos nodos para a transmissão de comandos MAC. O formato do quadro está representado na Figura 3.8. A Tabela 3.2 mostra os comandos MAC disponíveis.

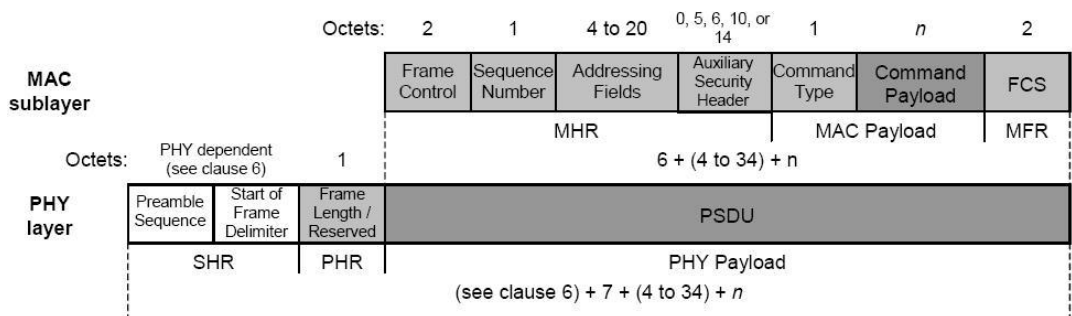


Figura 3.8: MAC command frame ([2]).

Tabela 3.2: Comandos MAC.

Command Frame Identifier	Command Name	RFD	
		Tx	Rx
0x01	Association Request	X	
0x02	Association Response		X
0x03	Disassociation Notification	X	X
0x04	Data Request	X	
0x05	PAN ID Conflict Notification	X	
0x06	Orphan Notification	X	
0x07	Beacon Request		
0x08	Coordinator Realignment		X
0x09	GTS Request		
0x0A – 0xFF	Reserved		

Fonte: [2]. p. 149

## 4 ARQUITETURA DO HARDWARE

Essencialmente, cada nodo da rede deve ter um rádio acoplado para permitir a transmissão e recepção de dados. Ele deve ser capaz de interpretar esses dados a fim de respondê-los ou de processá-los para executar alguma ação externa. Já que nas redes de automação residencial deseja-se fazer a integração de inúmeros componentes, é importante que cada um dos módulos seja simples, pequeno e barato.

Desde o início do projeto houve uma familiarização com uma certa arquitetura da empresa Freescale concebida para desenvolvimento de aplicações utilizando o padrão IEEE 802.15.4. Tal arquitetura inclui algumas interfaces desnecessárias para este projeto, ou pelo menos para a maior parte dele, como: porta RS-232, um acelerômetro para demonstrar o interfaceamento de um dispositivo externo com o micro-controlador, LEDs, botões e outras.

O hardware utilizado é fortemente baseado nesta arquitetura, porém com as devidas simplificações. Cada módulo é composto essencialmente por um micro-controlador, um circuito de rádio, uma antena, um LED e todos os pinos necessários para programá-lo e alimentá-lo (representação da Figura 4.1). Para este projeto foram utilizados dois tipos de módulos levemente distintos, que se diferenciam basicamente pela versão do circuito integrado de rádio e pelo micro-controlador que os mesmos utilizam. Cada uma deles está melhor especificado nos sub-itens subsequentes.

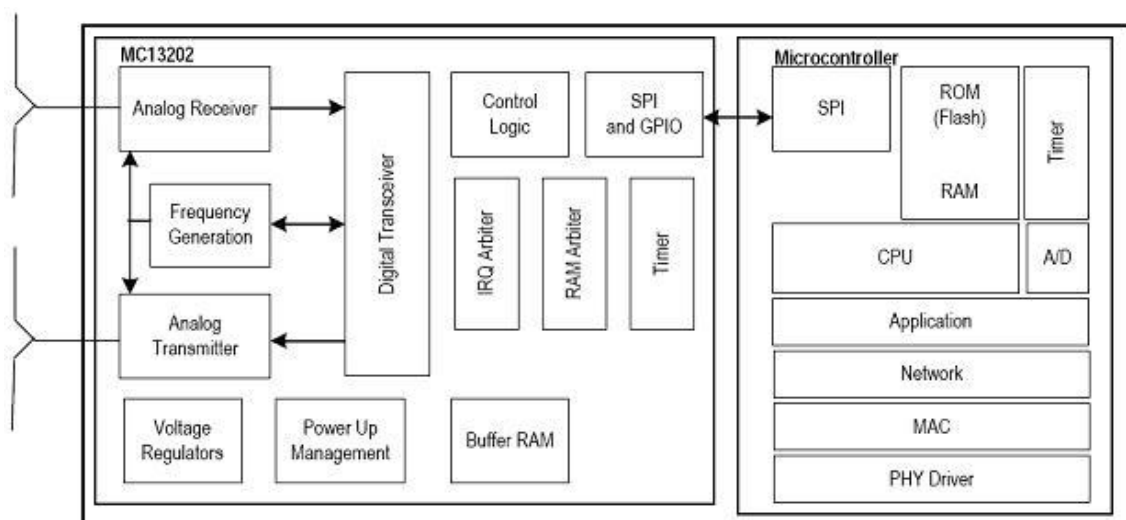


Figura 4.1: Diagrama em blocos em nível de sistema ([12]).

Ambos os módulos utilizam transceptores de RF fabricados pela empresa Freescale. Tais dispositivos de radio operam na banda de 2.4 GHz em um dos 16 canais



selecionáveis. Eles possuem uma taxa de transmissão no ar de 250 Kbps utilizando modulação O-QPSK e codificação DSSS. Além disso, têm compatibilidade com os softwares SMAC e PHY/MAC do padrão IEEE 802.15.4.

A interface do circuito de rádio com o micro-controlador é feita através de um SPI (Serial Peripheral Interface) de quatro fios. Isso permite o seu uso com uma vasta gama de diferentes micro-controladores.

#### 4.1 Placas de rádio do laboratório LAIS

O primeiro contato do autor com um circuito de rádio IEEE 802.15.4 micro-controlado, se deu através destes módulos já terem sido previamente utilizadas pelo grupo de pesquisa do laboratório LAIS (Laboratório de Automação e Integração de Sistemas) da UFRGS. O mesmo dispõe de dois módulos funcionais: o primeiro, a arquitetura mais complexa mencionada acima (Figura 4.2), e o segundo, a arquitetura simplificada (Figura 4.3). A partir do módulo de referência da empresa Freescale, a empresa AirWire criou os módulos de arquitetura simplificada.

Os módulos são idênticos quanto ao funcionamento do rádio através do micro-controlador. Ambos possuem: transceptor de RF MC13192 e o micro-controlador MC9S08GT60 da empresa Freescale. Este último pertence à linha de controladores HCS08 que têm como principais características o baixo custo e o alto desempenho. Algumas das características que tiveram que ser levadas em consideração no projeto:

- Total de memória FLASH: 60 Kbytes;
- Total de memória RAM: 4 Kbytes;
- Tamanho da palavra: 8 bits;
- Frequência do relógio: até 40 MHz.



Figura 4.2: Módulo de arquitetura complexa

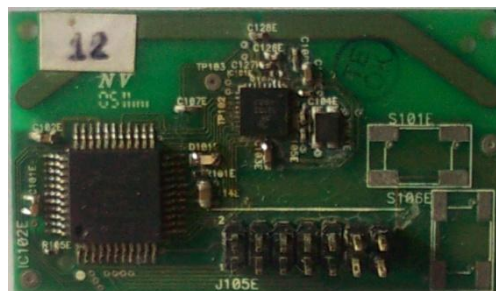


Figura 4.3: Módulo de arquitetura simplificada

## 4.2 Placas de rádio “Design Reference” da empresa Freescale

Em função do número reduzido de módulos, fator que impossibilitaria mais tarde um teste mais abrangente e exaustivo da rede, o autor decidiu criar novos módulos. O prévio conhecimento do autor das placas de rádio do laboratório LAIS, o levou inicialmente a tentar reproduzir tais placas. Isso não pôde ser feito devido à obsolescência de certos componentes eletrônicos.

Algumas alternativas foram consideradas e por fim foi tomada a decisão de se construir novas placas (semelhantes às outras) a partir de um novo “design reference” da empresa Freescale, o qual vem acompanhado dos leiautes do circuito impresso, juntamente com o diagrama de blocos do mesmo.

Considerando que o projeto envolve RF, tomar como base algo desenvolvido por profissionais do ramo é altamente recomendável. Em função disso, o autor decidiu construir as placas tal como estavam especificadas nos documentos da empresa Freescale. Os passos de fabricação adotados para a construção das placas está descrito no próximo item.

Os módulos do “design reference” são muito semelhantes aos do laboratório LAIS. Eles diferem basicamente na versão do CI de rádio e do micro-controlador que utilizam. Neste “design”, o transceptor RF utilizado é o MC13202 e o micro-controlador, o MC9S08QE128. O último também pertencente à linha HCS08 que, além das características de baixo custo e alto desempenho, também incorpora as de baixo consumo. Abaixo, estão listadas algumas características deste controlador:

- Total de memória FLASH: 128 Kbytes;
- Total de memória RAM: 8 Kbytes;
- Tamanho da palavra: 8 bits;
- Frequência do relógio: até 50 MHz.

### 4.2.1 Etapas de fabricação

1. Encomenda dos componentes principais.

A primeira encomenda feita foi dos circuitos integrados do rádio e do micro-controlador. Os mesmos foram pedidos como amostra grátis da empresa Freescale, que envia até seis componentes de cada tipo.

2. Busca e encomenda dos demais componentes eletrônicos.

Aqui começaram os primeiros problemas de seguir um “design reference” à risca. Todos os componentes passivos especificados pelo “design” são SMD (“Surface Mount Device”). Eles são extremamente difíceis de serem encontrados no mercado nacional. Por este motivo, tiveram que ser importados. No caso, todos os componentes restantes foram encomendados de uma empresa norte-americana que faz vendas através da internet, chamada Mouser Electronics.

### 3. Confeção do circuito impresso.

Este passo é o mais fácil e o menos demorado. Há várias empresas no mercado que fabricam placas de circuito impresso e não foi difícil o autor encontrar uma boa indicação de uma empresa. As placas de circuito impresso foram fabricadas pela empresa Stick, localizada no estado de Minas Gerais, que foi uma das poucas empresas que se disponibilizou a confeccionar um número reduzido das mesmas.

### 4. Solda dos componentes.

Este foi definitivamente o passo mais trabalhoso de todos. Além do problema da solda de componentes SMD que é muito cara para ser programada em uma máquina e muito difícil de ser soldada à mão (os menores componentes possuem dimensões de 4x2 mm), havia ainda o problema da solda do circuito integrado do rádio. O mesmo possui um encapsulamento extremamente complicado de ser soldado, pois seus “pads” se encontram embaixo do circuito.

Após uma busca que levou aproximadamente um mês e meio, através de uma indicação, foi encontrada uma pessoa que faz esse tipo de solda à mão para os protótipos de uma empresa localizada na região metropolitana de Porto Alegre.

Estas foram as etapas de fabricação adotadas pelo autor. As mesmas demandaram muito mais tempo para serem executadas devido à inexperiência nesta área. As placas foram fabricadas e estão representadas na Figura 4.4. Seus custos unitários estão descritos no próximo item.

Analisando o módulo, podemos observar mais uma diferença em relação aos demais, que é o formato da antena. Os módulos do laboratório LAIS possuem uma antena dipolo, cujo formato se assemelha a um ‘T’, enquanto que a antena dos módulos fabricados é uma antena-F, que tem esse nome devido ao seu formato se assemelhar a um ‘F’. Apesar de sua construção ser aparentemente mais complexa que a antena dipolo, as vantagens em desempenho devido ao uso de uma antena com o design da antena-F, fazem com que estas sejam mais comumente utilizadas ([15]), apesar de a antena dipolo ser de menor custo de implementação.



Figura 4.4: Módulo “design reference”.

#### 4.2.2 Custos do projeto

A Tabela 4.1 mostra os custos do projeto, incluindo toda a mão de obra e o custo unitário de cada componente eletrônico.

Tabela 4.1: Custos do projeto.

Descrição	Tamanho	Quantidade	Custo (Dólar)	Custo (Real)
Capacitor 1.8 pF	0402	6	0.02	0.04
Capacitor 8 pF	0402	12	0.02	0.04
Capacitor 0.1 uF	0402	42	0.03	0.04
Capacitor 10 pF	0402	12	0.02	0.04
Capacitor 10 uF	0805	12	2.13	3.79
Capacitor 1.0 uF	0805	6	0.07	0.12
LED	0603	6	0.12	0.21
Header	-	3	3.50	6.23
Indutor 3.3 nH	0402	12	0.90	1.60
Resistor 2 Kohm	0402	6	0.07	0.12
Resistor 0 ohm	0402	24	0.08	0.14
Resistor 100 Kohm	0402	6	0.07	0.12
Resistor 330 ohm	0402	6	0.07	0.12
Balun 50 to 50	-	6	0.38	0.67
MCU MC9S08QE128	-	6	-	-
Transceptor MC13202	-	6	-	-
Regulador de Tensão 5-3.3 V	-	6	1.08	1.92
Oscilador	-	6	0.61	1.09
Placa de circuito impresso	-	6	2.25	4.00
Solda	-	6	11.24	20.00
<b>Total</b>		195	146.11	260.08

O custo total, considerando a confecção de seis placas, foi de R\$ 260,08. Ou seja, o preço unitário de cada placa é aproximadamente R\$ 43,00.

## 5 ARQUITETURA DO SOFTWARE

O software é a parte mais extensa deste projeto. Foram utilizados cinco diferentes tipos de softwares essenciais ao projeto, sendo que somente o primeiro da lista a seguir não foi desenvolvido pelo autor. São eles:

- API do padrão IEEE 802.15.4;
- Software do coordenador PAN;
- Software dos nodos;
- Software para interface entre: “web” e comunicação RS-232;
- CGI que gera o HTML da interface “web”.

Cada um dos programas será detalhado adiante. Inicialmente, faremos uma análise geral do funcionamento de todos em conjunto a fim de deixar clara a utilidade e necessidade de cada um deles. Podemos visualizar a estrutura geral do sistema através da Figura 5.1.

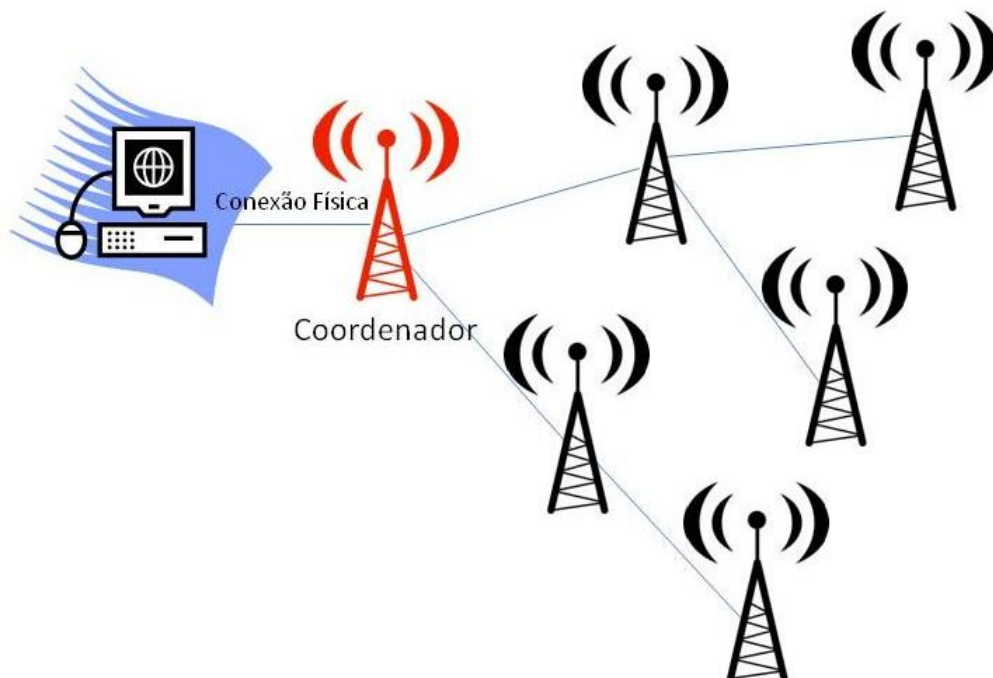


Figura 5.1: Estrutura do sistema.

Tomamos como ponto de partida os nodos da rede. Cada um deles, sendo o coordenador ou não, deve possuir a mesma interface de acesso ao meio, a qual é

definida pelo padrão IEEE 802.15.4. Neste sistema, o padrão está implementado através de uma API, que é utilizada pelos softwares do coordenador PAN e dos nodos.

Estes softwares, por sua vez, são responsáveis pelo controle da transmissão e recebimento de mensagens na rede. Suas diferenças serão analisadas a seguir, quando cada um deles for melhor detalhado.

As mensagens de controle, que disparam ações nos nodos são provenientes do PC. É através da interface “web” que o usuário tem controle dos nodos da rede. A comunicação entre a interface “web” e o coordenador de rede é feita por meio de um software que se comunica com o coordenador através da porta RS-232 a partir de uma mensagem gerada pelo CGI da interface.

Podemos ter uma visão macro do funcionamento em conjunto de todos os softwares que compõem o projeto, através da Figura 5.2.

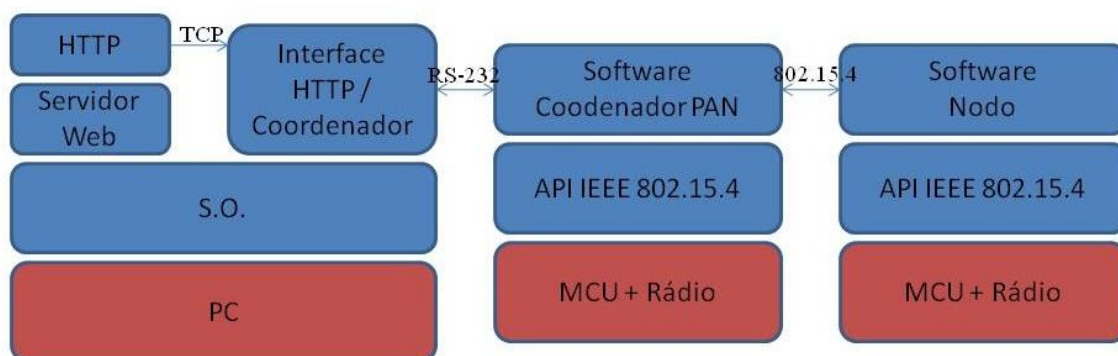


Figura 5.2: Diagrama em blocos da comunicação dos softwares.

## 5.1 Ambiente de desenvolvimento

Para iniciar a codificação dos softwares do coordenador e dos nodos que implementam a rede, foi necessário montar todo o ambiente para desenvolvimento de aplicações para a MCU utilizada. Este ambiente inclui:

- IDE CodeWarrior;
- Driver para utilização do OpenSource BDM;
- API IEEE 802.15.4.

### 5.1.1 CodeWarrior

O CodeWarrior (Figura 5.3) é uma IDE mantida e comercializada pela empresa Freescale, disponível para diversos sistemas operacionais, que provê ao usuário todo o ferramental necessário para desenvolver aplicações embarcadas complexas. Além de incluir o “cross-compiler” necessário para gerar código para a arquitetura HCS08 utilizada, ele também possui uma interface para “debug” geralmente encontrada em sistemas embarcados, chamada BDM.

Ela foi escolhida para ser utilizada pois, além de possuir um bom compilador para as linguagens C e C++, em sua versão mais nova permite o uso do BDM através da porta USB. O BDM além de ser utilizado como interface de “debug” padrão, ele também pode ser utilizado para realizar a programação da memória FLASH das placas, funcionalidade que também vem incorporada à ferramenta.

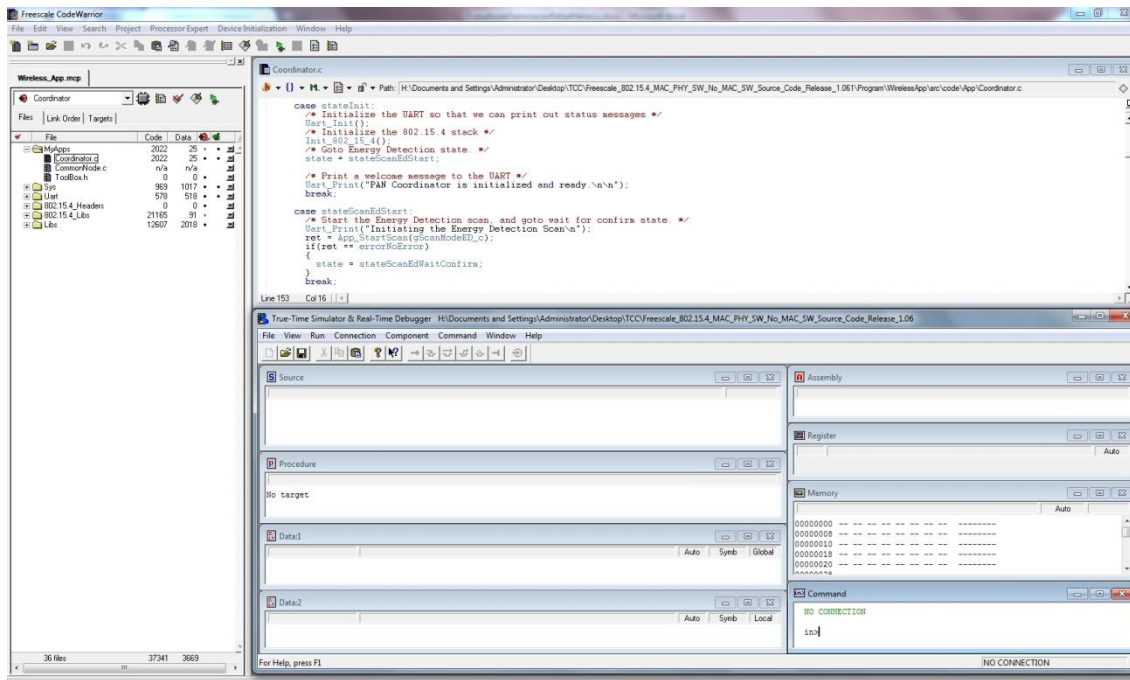


Figura 5.3: CodeWarrior IDE.

A conexão da interface BDM com o computador é feita por meio de um circuito micro-controlado adicional, chamado de OpenSource BDM, que tem por objetivo implementar a comunicação via USB do computador com qualquer dispositivo que utilize o BDM, de maneira transparente. Há leiautes gratuitos e de fácil acesso deste circuito na internet ([16]). Além disso, para utilizá-lo deve-se instalar um driver que é disponibilizado apenas para o sistema operacional Windows XP. A Figura 5.4 representa esta placa.



Figura 5.4: Circuito OpenSource BDM

### 5.1.2 API IEEE 802.15.4

Uma evolução em relação aos objetivos iniciais do projeto é o uso da API do padrão IEEE 802.15.4. Em sua especificação inicial, o projeto previa somente o uso de uma pilha de código chamada SMAC, que implementa o mínimo necessário para fazer alguma comunicação no ar, utilizando os transceptores da empresa Freescale. Essa escolha permitiu ao projeto tomar uma forma mais complexa em alguns quesitos que anteriormente seriam deixados de lado em função do custo em tempo que levariam para serem implementados. Tais quesitos envolvem principalmente: detecção de um canal não ocupado, detecção de colisão na hora da transmissão, uso do mecanismo de

“backoff”, identificação do nodo com a melhor qualidade de “link” em uma determinada vizinhança, entre outros.

Esta API implementa todo o software do 802.15.4 MAC/PHY, tal como definido pela norma. Seu uso é relativamente simples e é feito através de chamadas de sub-rotinas. Ela define basicamente dois tipos diferentes de dispositivos e três tipos de interfaces que são acessíveis a partir da aplicação (Figura 5.5), descritos nos itens a seguir.

O acesso a essas interfaces é feito por meio de envio de mensagens, através de uma mesma chamada *MSG\_Send(NWK\_MLME, pMsg)*, por exemplo, na qual é especificada a interface e a mensagem a ser interpretada por ela: *NWK\_MLME* e *pMsg*, respectivamente.

#### 5.1.2.1 Dispositivo de funções reduzidas

Dispositivo que inclui apenas um subconjunto das funcionalidades do IEEE 802.15.4. Este dispositivo pode atuar somente como um nó folha na rede, não tendo capacidade para fazer roteamento de pacotes. É normalmente utilizado por dispositivos que têm restrições de consumo de energia.

#### 5.1.2.2 Dispositivo de funções completas

Dispositivo que contém todas as funcionalidades do IEEE 802.15.4. Pode atuar tanto como um nó folha, como pode fazer roteamento de pacotes. Normalmente utilizado por dispositivos que tenham uma fonte de alimentação abundante. Por ser mais complexo, seu código é maior e ele ocupa mais espaço em memória.

Neste projeto, todos os nodos são definidos como dispositivos deste tipo.

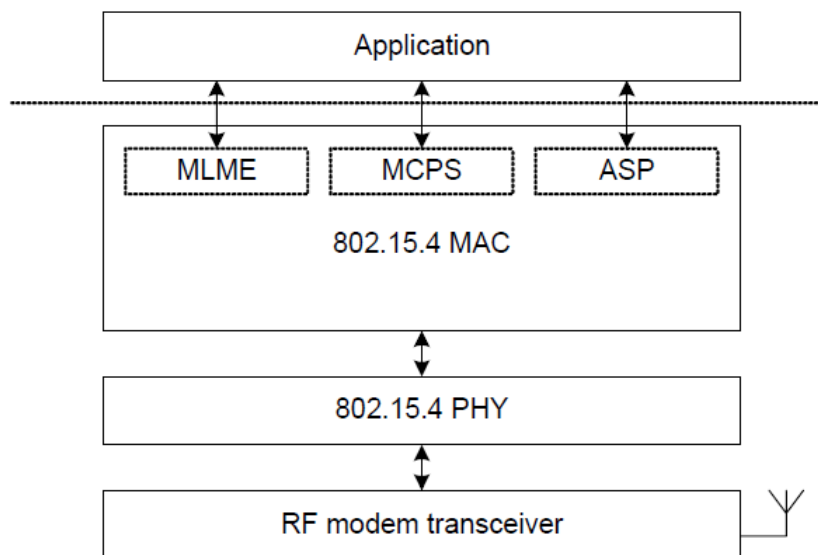


Figura 5.5: Diagrama em blocos da API IEEE 802.15.4 ([14]).

#### 5.1.2.3 Interface MLME

Interface utilizada por todos os comandos MAC do padrão IEEE 802.15.4. É utilizada, por exemplo, pela aplicação para fazer requisições de associação na rede e receber confirmação da associação. Esta interface é definida na especificação do padrão.



#### 5.1.2.4 Interface MCPS

Interface utilizada por todas as primitivas relativas a dados do padrão IEEE 802.15.4. A aplicação deve utilizar esta interface para todas as operações de envio e recebimento de dados. Esta interface também é definida na especificação do padrão.

#### 5.1.2.5 Interface ASP

Interface utilizada por várias funcionalidades de suporte à aplicação. Por exemplo, para a aplicação dizer ao hardware para entrar em modo de baixo consumo. Esta interface não é utilizada neste projeto e não é definida pelo padrão.

### 5.1.3 Porte da API

A API IEEE 802.15.4 foi projetada para funcionar com micro-controladores da família HCS08 e transceptores RF da família 1320x. Entretanto, apesar de os controladores serem similares, seus mapas de memória são completamente diferentes.

Esta parte, apesar de ser um trabalho fácil, é cansativa, pois a comparação entre os mapas de memória deve ser feita entrada a entrada, e nem sempre o registrador mapeado possui o mesmo nome. Além disso, o mapa é consideravelmente longo, tendo aproximadamente cento e trinta entradas. Nos casos em que os registradores mapeados possuem nomes diferentes, sua funcionalidade teve que ser observada para achar o equivalente no outro micro-controlador.

Abaixo segue um exemplo da mudança de um trecho do mapa de memória:

```
#define SCI1BDH (*(volatile unsigned char*)(0x20))
#define SCI1BDL (*(volatile unsigned char*)(0x21))
#define SCI1C1 (*(volatile unsigned char*)(0x22))
#define SCI1C2 (*(volatile unsigned char*)(0x23))
```

```
#define SCI1BDH (*(volatile unsigned char*)(0x18))
#define SCI1BDL (*(volatile unsigned char*)(0x19))
#define SCI1C1 (*(volatile unsigned char*)(0x1A))
#define SCI1C2 (*(volatile unsigned char*)(0x1B))
```

## 5.2 Software dos módulos

O principal foco deste projeto se encontra no software dos módulos. O mesmo ocupa exatamente a função da camada de aplicação que podemos visualizar através da Figura 5.5 e é o responsável pela implementação da rede em si. Como podemos ver pela mesma figura, ele funciona por meio de chamadas de sub-rotinas da API IEEE 802.15.4, que por sua vez, já implementa todo o controle de hardware necessário.

Os softwares do coordenador PAN e dos nodos comuns são levemente diferentes. O coordenador, por ser o mestre da rede, deve possuir algumas funcionalidades a mais que os nodos comuns. São elas:

- Definição do canal que será utilizado pela rede;
- Atribuição de endereços para os nodos da rede;
- Conhecimento de todos os nodos da rede, relacionando eles com um dado nome pelo qual são conhecidos pelo servidor “web”;
- Separação das mensagens individuais para cada nodo;
- Comunicação RS-232 para execução de comandos.

Antes de iniciar a descrição detalhada dos softwares dos módulos, devemos descrever o funcionamento geral da rede.

Previendo uma futura melhoria da rede no quesito de consumo de potência, o autor decidiu que a melhor forma de implementar a mesma, seria através de um sistema em que os nodos fazem “polling” nos seus coordenadores a fim de verificar se os mesmos possuem mensagens para eles. Essa decisão foi tomada devido a dois fatores determinantes: vantagem para nodos com necessidade de baixo consumo de potência e custo de implementação.

Baixo consumo de potência e custo de implementação estão intimamente relacionados, pois outra forma de se alcançar o baixo consumo é através da criação de uma rede sinalizada. Na rede sinalizada, os coordenadores transmitem quadros de sinalização em tempos determinados, e através de um mecanismo de sincronismo, os nodos se ativam quando esses quadros são esperados e verificam se o seu respectivo coordenador tem alguma mensagem para eles. Na implementação por “polling”, de tempos em tempos o nodo acorda e pergunta diretamente ao seu coordenador se ele possui alguma mensagem para o nodo. A rede sinalizada possui uma vantagem de potência em relação à de “polling”, pois o custo energético de uma transmissão no meio é muito maior do que o custo de uma recepção. Em contrapartida, e daí a decisão desta construção, a implementação de uma rede sinalizada é extremamente mais complexa do que uma rede que opera por “polling”.

A futura melhoria na rede acima mencionada vem do fato de que futuramente pode-se querer um nodo cuja fonte de alimentação é uma bateria de baixa capacidade de armazenamento energético. Para este nodo, a qualidade de poder ser um coordenador e retransmitir pacotes não é necessária e nem desejada, já que fazendo isso, seu consumo de energia seria mais elevado.

Visando futuramente poder inserir esse tipo de nodo sem precisar alterar a estrutura geral da rede, a rede criada possui o seguinte formato: supondo que há um nodo ‘A’ já associado ao coordenador PAN e um novo nodo ‘B’ deseja se associar ao nodo ‘A’, o fluxo de mensagens que deverá trafegar na rede é como representado na Figura 5.6.



Figura 5.6: Mensagens de associação.

1. O nodo 'B' faz uma varredura na rede para descobrir os coordenadores à sua volta.
2. O coordenador com o "link" de melhor qualidade é escolhido e então o nodo 'B' envia a ele uma mensagem de associação (no caso, o nodo 'A').
3. O nodo 'A' envia uma mensagem de associação ao seu coordenador (que não precisa necessariamente ser o coordenador PAN), como se ele mesmo quisesse se associar.
4. O coordenador do nodo 'A' envia uma resposta da associação ao nodo 'A', com o endereço que o mesmo terá na rede (a cadeia segue até o PAN, pois somente ele atribui endereços na rede).
5. O nodo 'A' atualiza sua tabela de nodos conectados, cadastrando o endereço do nodo 'B'.
6. O nodo 'A' envia uma resposta da associação ao nodo 'B', informando-o de seu endereço.

A partir do procedimento de associação visto acima, podemos inferir três coisas essenciais para o entendimento da rede:

- Um coordenador pode ter vários nodos associados a ele, sem que necessariamente ele esteja conectado diretamente a todos eles;
- Cada coordenador deve fazer "polling" no seu coordenador por todos os nodos que estão associados a ele (mecanismo descrito mais adiante);
- O coordenador PAN tem conhecimento de todos os nodos da rede, sem saber necessariamente onde eles estão.

A primeira coisa que podemos pensar ao entender o segundo item listado acima, é que em uma rede grande, um coordenador que estiver próximo ao PAN, poderá ter um número muito grande de nodos associados a ele, e ele poderá se tornar um gargalo. Como está implementada no projeto, a estrutura não prevê nenhuma forma de contornar este problema, exceto no bom senso do usuário em espalhar bem os nodos de forma que isso não ocorra.

### **5.2.1 Coordenador PAN**

Conforme mencionado anteriormente, este módulo é responsável pelo controle e manutenção da rede. Por este motivo, seu software é mais complexo do que o software dos nodos comuns e é válido mencioná-lo primeiro, já que seu entendimento facilitará bastante o entendimento do software dos demais nodos.

Começaremos descrevendo o processo de inicialização do coordenador para depois abordarmos especificamente o funcionamento da parte do software que faz o controle da rede.

#### *5.2.1.1 Inicialização do coordenador*

A inicialização é feita através de uma máquina de estados que realiza as funções a seguir nesta mesma ordem que aparecem:

1. Inicialização da porta serial.
2. Inicialização da pilha 802.15.4.

3. Varredura de detecção de energia. Cada um dos dezesseis canais da banda de 2.4 GHz é monitorado por um tempo fixo e retornado o seu valor de energia. Este valor varia em passos de  $\frac{1}{2}$  dBm, com 0 representando -80 dBm (valor mínimo teórico) e 160 representando 0 dBm (valor máximo teórico). O canal com o menor nível de energia é o escolhido.
4. Identificação como coordenador PAN. Diz para o MLME que este é o coordenador da rede iniciando uma PAN com identificação *0xBEEF*, que seu endereço é *0xCAFE*, e habilitando outros “flags” de controle.
5. Entrada no estado de escuta, que é o laço principal do sistema.

Após todo o processo de inicialização, o sistema fica em um laço infinito que faz basicamente três verificações a cada iteração: chegada de mensagens através da porta RS-232 e chegada de mensagens através das interfaces MCPS e MLME. Isso define a entrada na parte de controle da rede que está descrita a seguir.

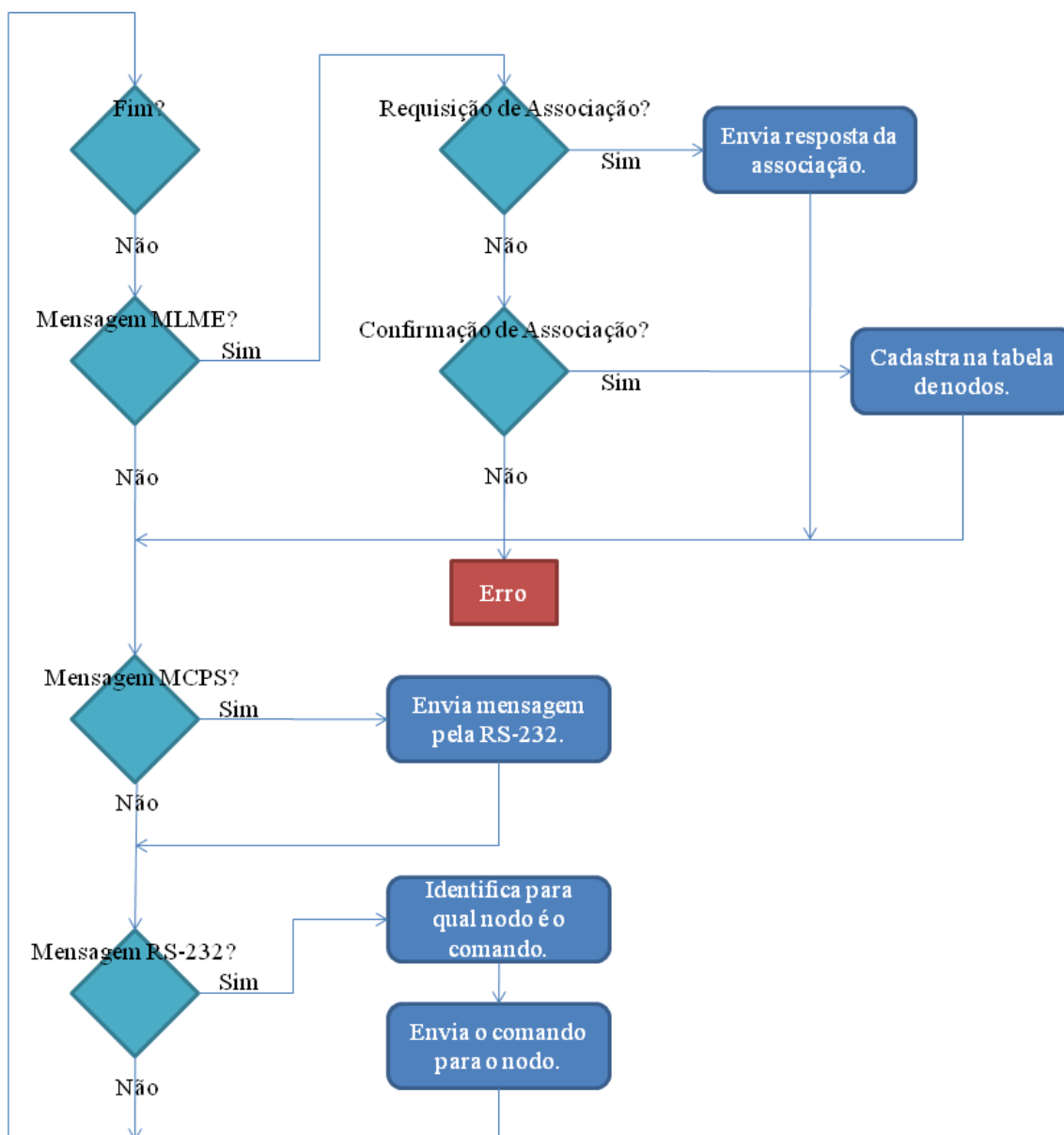


Figura 5.7: Fluxograma do laço principal do coordenador PAN.

Como podemos observar a partir da Figura 5.7, na qual está representado o fluxograma de controle da rede do coordenador PAN, podemos ver que há uma prioridade no tratamento de mensagens da interface MLME, seguido da interface MCPS e por último da porta RS-232. Isso ocorre devido às mensagens da primeira interface serem referentes ao controle do estado da rede.

Nesta fase, o sistema ao identificar o recebimento de uma mensagem, deve realizar algumas verificações e em seguida processá-la primeiramente de acordo com a interface pela qual ela veio e depois, com o seu tipo. O processamento dado para cada mensagem está descrito nos itens a seguir:

#### *5.2.1.2 Mensagem proveniente da interface MLME*

Sendo uma mensagem proveniente desta interface, ela pode servir para dois propósitos: o primeiro significa que algum dispositivo está querendo se associar ao coordenador PAN, e o segundo ocorre quando há confirmação de que a resposta a uma associação foi enviada com êxito.

No primeiro caso, o coordenador PAN calcula um novo endereço que ainda não foi utilizado na rede (os endereços começam com numeração *0x0001*, e aumentam em uma unidade a cada associação) e logo em seguida, envia a mensagem de resposta à associação. Vale ressaltar que durante a associação não ocorre o mecanismo de “polling”, significando que o rádio do dispositivo que está requisitando a associação permanece ligado durante todo o processo.

No segundo caso, o coordenador PAN que está recebendo a confirmação da associação, simplesmente cadastra aquele dispositivo na tabela de nodos associados, e a partir deste momento, o mesmo está fazendo parte da rede.

#### *5.2.1.3 Mensagem proveniente da interface MCPS*

Neste projeto não está definida nenhuma utilidade para comunicação de dados provenientes de um nodo para o coordenador PAN, apesar de a rede suportar este tipo de comunicação. Um exemplo de aplicação em que é interessante o envio de mensagens ao coordenador PAN, é no caso de um sensor que de tempos em tempos transmite dados, seja para informar o usuário, seja para esperar alguma ação dele.

Como o foco do projeto não é dar exemplos de uso para todas as possíveis situações em que podemos colocar a rede, e sim a definição de uma rede que dê suporte a todos os tipos de aplicação, a única tarefa que o coordenador PAN realiza ao receber uma mensagem de dados, é transmiti-la diretamente pela porta RS-232.

#### *5.2.1.4 Mensagem proveniente da porta RS-232*

A mensagem recebida através da porta RS-232 é idêntica à “string” gerada pelo CGI que opera no servidor “web” (a mesma é especificada mais adiante neste documento). Cabe ao coordenador PAN dividir a mensagem em pedaços a fim de identificar se ela se refere a um único comando destinado a somente um nodo, ou se ela se refere a múltiplos comandos para diferentes nodos. Além disso, o coordenador PAN também tem como tarefa associar o nome proveniente da mensagem enviada pelo CGI ao endereço do nodo ao qual esta mensagem está destinada.

O conteúdo da mensagem para cada nodo pode ser diferenciado, pois cabe somente ao nodo saber interpretá-lo e, no caso, ao coordenador PAN para saber como deve construí-la. Pensando nisso, o autor criou uma regra simples para a construção destas mensagens, que está descrita no item 5.4, na página 42.

### 5.2.2 Nodos genéricos

A compreensão do funcionamento do coordenador PAN já deve ter levado o leitor a ter uma noção de como deve ser o funcionamento básico dos nodos comuns da rede.

O nodo genérico, tal como foi implementado neste projeto, possui a habilidade de ser um coordenador. Isso significa que ele pode receber associações de outros nodos a fim de retransmitir pacotes. Como já foi mencionado no item 5.2 da página 33, neste escopo, um nodo comum não pode assumir o papel de um nodo folha abdicando assim da sua capacidade de retransmissão de pacotes, pois este não está definido para tal.

Analogamente à análise do coordenador PAN, iniciaremos descrevendo o processo de inicialização do nodo comum, para depois entrarmos nos detalhes de implementação de como ele age na rede.

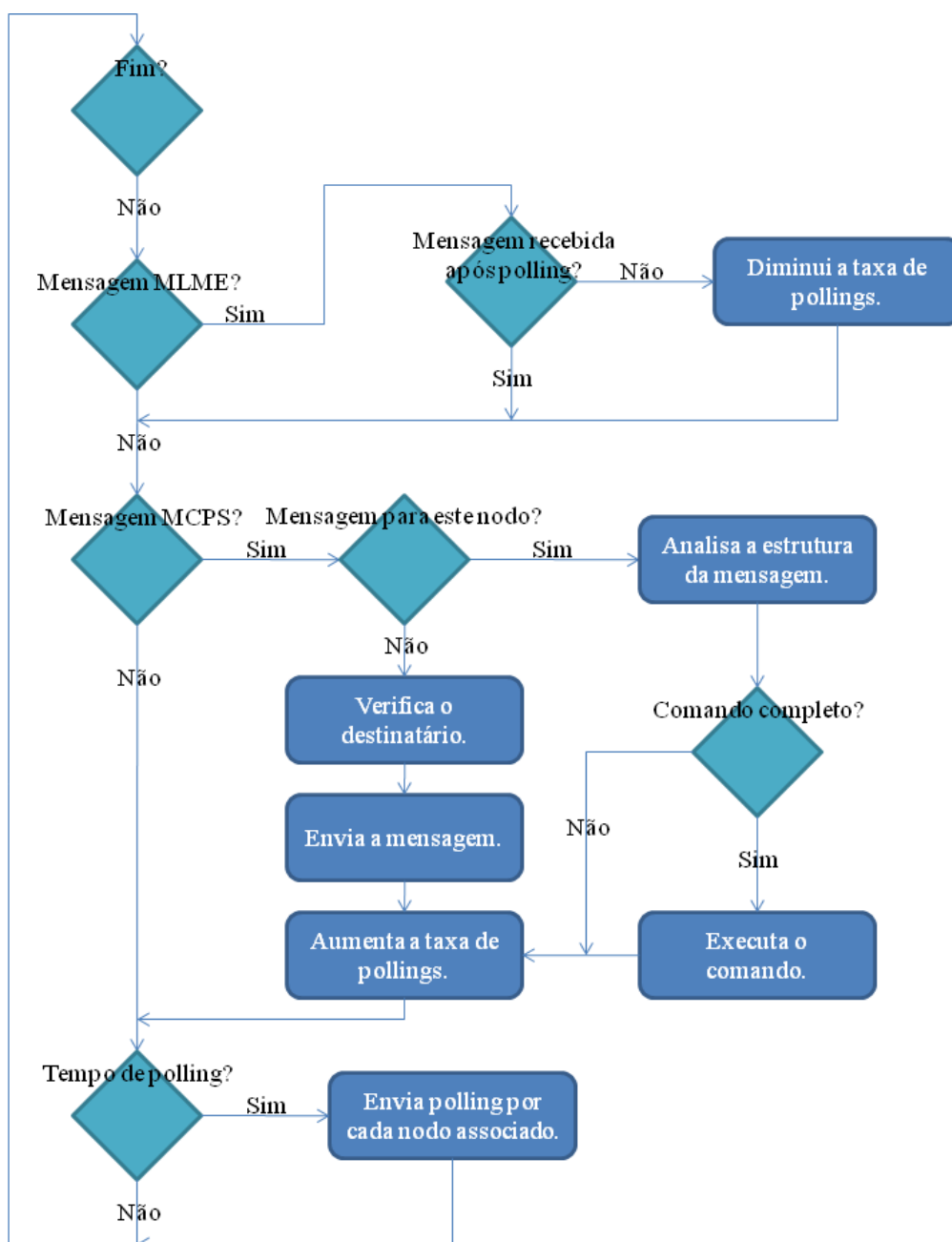


Figura 5.8: Fluxograma do laço principal do nodo genérico.

### 5.2.2.1 Inicialização do nodo genérico

1. Inicialização da porta serial.
2. Inicialização da pilha 802.15.4.
3. Varredura de coordenadores vizinhos. Uma mensagem de busca é enviada à interface MLME, que por sua vez envia mensagens consultando os coordenadores vizinhos na vizinhança do nodo. A interface MLME retorna uma lista de coordenadores, com suas respectivas energias de sinal. O coordenador com o sinal de maior energia é o escolhido.
4. Associação ao coordenador escolhido.
5. Entrada no estado de escuta, que é o laço principal do sistema.

Quando a inicialização é concluída, o sistema entra em um laço infinito que basicamente deve fazer duas verificações obrigatórias de mensagens nas interfaces MLME e MCPS, e depois fica livre para interagir com o dispositivo controlado, se for o caso. Esse bloco de interação, se fosse necessário no nodo implementado para demonstração do projeto, deveria ser posicionado logo abaixo do teste de verificação de chegada do tempo de “polling”, visualizado no fluxograma de execução do nodo comum representado na Figura 5.8.

### 5.2.2.2 Mensagem proveniente da interface MLME

No caso do software do nodo genérico, a única mensagem que pode chegar a partir desta interface, é a indicação de que a interface MCPS recebeu ou não alguma mensagem de dados após fazer um “polling” no seu coordenador de rede.

Esta mensagem é particularmente útil, pois com ela é possível controlar a taxa de “pollings” gerados no coordenador. Caso tenhamos recebido uma mensagem de dados, a chance de recebermos outra logo em seguida é alta, pois pela própria característica do problema que estamos abordando, mensagens para os nodos não são frequentes. Apesar disso, quando a central está se comunicando com um determinado nodo, essa comunicação pode ocorrer por múltiplas mensagens, e é interessante que neste caso, a taxa de “pollings” seja maior.

Ao receber a indicação da interface MLME de que um “polling” não obteve sucesso, ou seja, que o mesmo não encontrou uma mensagem para o nodo no seu coordenador, apenas reduz-se a taxa de “pollings”.

Essa redução não é gradativa. Quando a taxa é reduzida, significa que fixamos o seu período em *820 ms*, e quando é novamente aumentada, o seu período é fixado em *82 ms*, ou seja, a taxa é dez vezes mais rápida. O aumento da taxa de “pollings” ocorre durante o recebimento de outro tipo de mensagem, visto a seguir.

### 5.2.2.3 Mensagem proveniente da interface MCPS

No recebimento de uma mensagem da interface MCPS, devemos verificar inicialmente se o pacote está destinado para este nodo ou não. Caso o pacote não se destine a este nodo, devemos transmitir o pacote para o nodo ao qual ele está destinado, tal como ele está. Vale lembrar que a transmissão do pacote pela API não ocorre instantaneamente; ao contrário, ela coloca a mensagem em uma tabela de mensagens que só serão enviadas quando os nodos aos quais elas se destinam fizerem um “polling” neste coordenador.

Caso a mensagem esteja destinada a este nodo, devemos então verificar se o pacote que recebemos representa uma mensagem completa ou não. Para tal, a cada nova mensagem recebida, é utilizada uma rotina que verifica o formato da mesma (descrito no item 5.4, página 42) e determina se ela está completa ou não. Caso não esteja completa, simplesmente concatenamos com a anterior (se existir) e esperamos um novo pacote. Após o recebimento de cada pacote, aumenta-se a taxa de “pollings” para o seu nível mais alto.

Ao obter uma mensagem completa, deve-se analisá-la a fim de determinar o seu significado. Para tanto, deve-se procurar na mesma a ação do nodo ao qual ela está se referindo e, em seguida, chamar a rotina que deve executar esta ação levando em consideração os demais parâmetros de controle também contidos nos dados. A implementação disso é trivial, considerando que a ação pode ser representada por um nome.

Com isso, pode-se observar que cada nodo deve conter sub-rotinas para cada ação que ele pode executar. No exemplo utilizado neste projeto, no qual acendemos um LED através da rede, determinamos que o comando que representa a ação de acendê-lo ou apagá-lo contém:

- Nome da ação: D1, simbolizando o LED que está mapeado diretamente no pino do micro-controlador que está ligado ao mesmo.
- Controle da ação: ‘N’ ou ‘F’, simbolizando que pode ser um comando de acender (ON) ou apagar o LED (OFF).

#### 5.2.2.4 Tempo de “polling”

Quando o tempo de “polling” é atingido, é necessário enviar uma mensagem de “polling” ao coordenador deste nodo. A mensagem de “polling” não se limita a uma única mensagem, caso este nodo tenha mais nodos associados a ele. Neste caso devemos enviar ao coordenador, uma mensagem de “polling” para cada um dos nodos associados. Isso é feito da seguinte forma:

- Para cada um dos nodos associados, devemos assumir o endereço dele;
- Logo após, devemos enviar a mensagem de “polling” ao nosso coordenador, como se fossemos aquele nodo.

Neste ponto todo o funcionamento da rede foi detalhado.

Agora, iremos analisar a forma com a qual a rede, ou melhor, o coordenador PAN se relaciona com os softwares restantes a fim de compor o que no início deste documento foi chamado de central.

### 5.3 Software de interface, e comunicação com o coordenador PAN

O software de interface, conforme mencionado anteriormente, é a página “web”, que tem por objetivo prover uma forma simples e prática para que o usuário possa controlar a rede. A interface deve, de alguma forma, se comunicar com o coordenador PAN a fim de informá-lo das ações que o usuário está realizando na rede. O uso de CGIs é a alternativa que foi usada para estabelecer esta comunicação. Porém, ela sozinha não implementa completamente o que precisamos.

O CGI, após ser executado, é encerrado. Isso não é interessante, pois neste caso desejamos manter uma comunicação constante com o coordenador PAN, a fim de exibir



as mensagens que o mesmo envia (única atitude tomada referente a estas mensagens nesta implementação), ou futuramente, trocar constantemente informações sobre o estado dos nodos. Por esta razão, sentiu-se a necessidade da criação de um software intermediário, com o qual a interface pudesse se comunicar e que ao mesmo tempo mantivesse ativa a comunicação com o coordenador PAN.

### 5.3.1 Software de comunicação entre a interface e o coordenador PAN

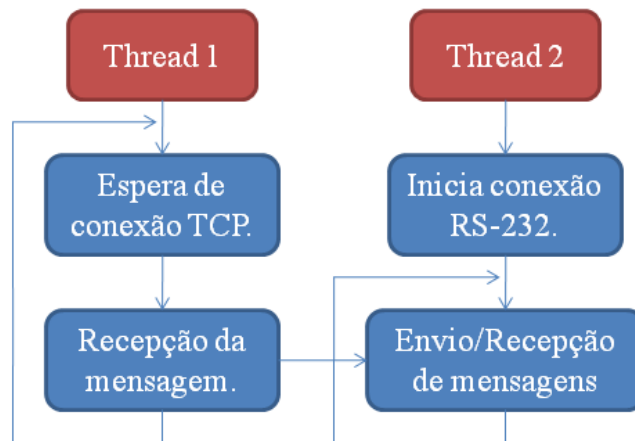


Figura 5.9: Diagrama em blocos do software intermediário.

Este software funciona conforme o diagrama em blocos representado pela Figura 5.9. Como podemos observar, ele é composto por duas threads principais que efetuam separadamente o controle das conexões com o CGI e com o coordenador PAN.

A primeira thread, ao ser criada, inicializa um “socket” que escuta através da porta 10101 e fica bloqueada na mesma aguardando conexões. A segunda thread por sua vez, abre a conexão RS-232 com o coordenador PAN e fica em um modo de escuta, simplesmente imprimindo na tela tudo que é enviado pelo coordenador.

Quando o aplicativo CGI abre uma conexão TCP com a primeira thread, a mesma imediatamente fica aguardando o recebimento de uma mensagem. Quando esta mensagem é recebida, a primeira thread coloca-a em uma área de memória compartilhada do programa, e habilita um “flag” que é constantemente monitorado pela segunda thread, e que corresponde a uma indicação de mensagem para ela. Logo após esta recepção de dados, a primeira thread imediatamente encerra a conexão e volta para o estado de escuta.

Quando a segunda thread fizer a verificação do “flag” de indicação de mensagens, ela imediatamente transmitirá a mensagem, tal como ela está, através da porta RS-232 para o servidor PAN e, obviamente, desabilitará o “flag”.

A construção deste software é bastante simples e envolve o uso de threads, mutexes, sockets e manipulação de arquivos para utilização da porta RS-232.

### 5.3.2 Software de interface “web”

Este software funciona através de uma página HTML, a qual precisa estar vinculada a um servidor “web” com recursos de CGI habilitados. A página é composta por botões que devem ser marcados a fim de se executar determinadas funções nos nodos, que no caso desta implementação, se resumem a acender e apagar LEDs.

Quando o botão de envio é pressionado, o servidor “web” gera uma string contendo informações sobre as marcações da página e inicia o aplicativo CGI. Este, por sua vez,

captura essa string, abre uma conexão TCP com o software intermediário e transmite a mesma. Após, ele redesenha a mesma página HTML e encerra, estando assim a página pronta para um novo evento do usuário. A página “web” está representada na Figura 5.10.

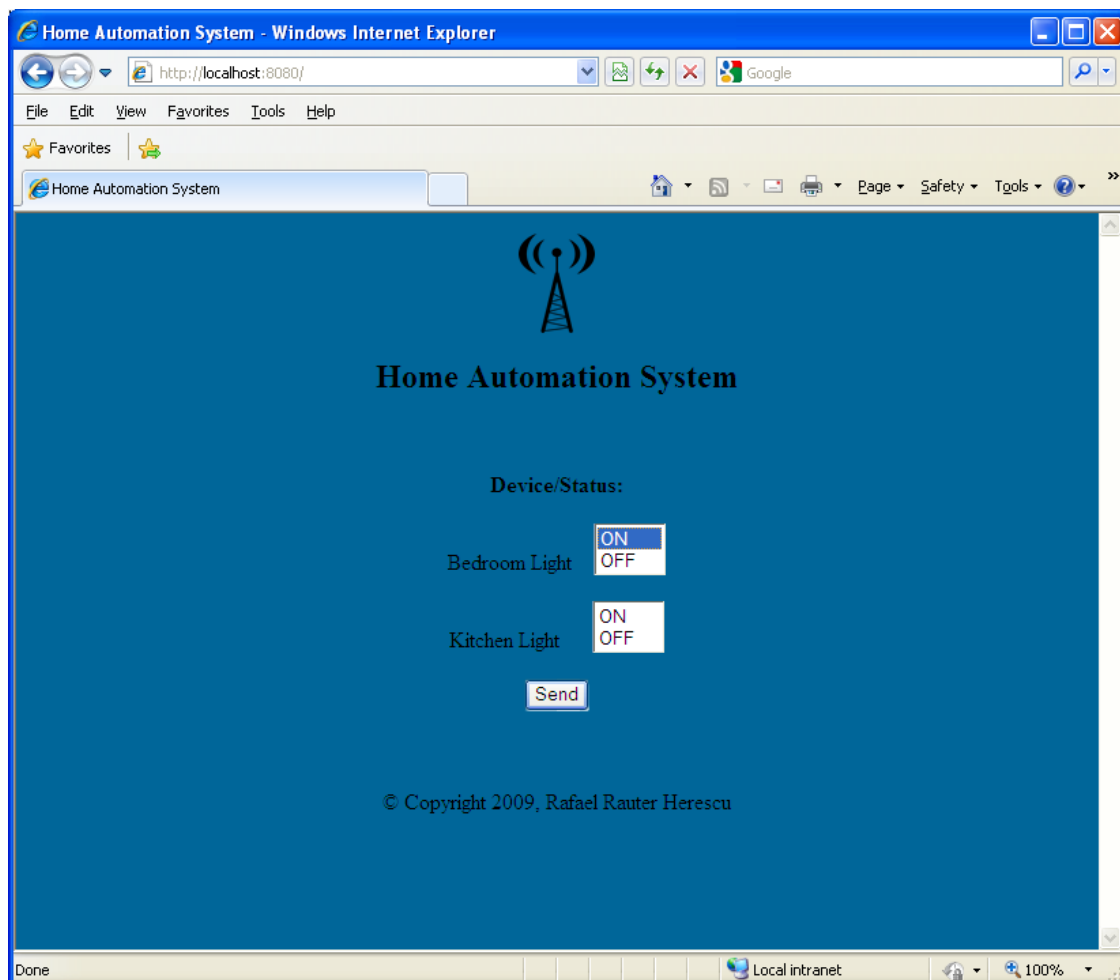


Figura 5.10: Página “web”.

#### 5.4 Regra de construção das mensagens

A regra de construção de mensagens é simples. Partindo-se da idéia de que dependendo do tipo de nodo que é controlado, pode-se querer trocar mensagens com qualquer tipo e número de caracteres, o autor pensou em um sistema de fácil implementação para codificação e decodificação das mensagens.

O sistema consiste basicamente em adicionar, no final da mensagem, dois caracteres que tem por objetivo indicar o fim da mensagem. São eles:

/X

Porém, considerando o problema de que se a mensagem a ser transmitida precisar conter exatamente essa estrutura, foi definido que a cada caractere ‘|’ que se deseja transmitir como dados da mensagem, deve-se adicionar mais um caractere ‘|’. Portanto caso for desejada a transmissão de uma mensagem que, por exemplo, é assim: “Testando caracteres |X”, a mensagem que será efetivamente construída e enviada é:

*Testando caracteres ||X|X*

## 6 TESTE E VALIDAÇÃO

O objetivo da fabricação de mais módulos além dos dois disponíveis inicialmente no laboratório LAIS era justamente poder testar a rede de uma forma otimizada. A escolha de fabricar um leiaute de “design reference” da empresa Freescale à risca levou o autor a produzir uma condição indesejada no projeto.

Como a confecção das novas placas começou a ser feita antes mesmo da codificação dos softwares da rede e do próprio uso e estudo da API IEEE 802.15.4, não foi possível perceber de antemão a dificuldade que iria surgir quando se fosse fazer o porte desta API para o micro-controlador utilizado no leiaute escolhido. O porte da API para o micro-controlador MC9S08GT60 (utilizado nas placas do laboratório LAIS), mencionado no item 5.1.3 da página 33, foi razoavelmente simples pelos motivos citados anteriormente. Entretanto, as diferenças entre o mapa de memória do micro-controlador para qual a API foi desenvolvida e o mapa do MC9S08QE128 (utilizado nas placas fabricadas pelo autor), são muito grandes.

Não somente os endereços do mapa de memória são diferentes, mas também os registradores mapeados por ele. A conversão do mapa de memória para este novo micro-controlador seria extremamente difícil pelo fato de que nem todos os registradores utilizados pela API possuem um correspondente direto neste novo micro-controlador. A conversão é possível de ser feita, pois o estudo que o autor acabou fazendo para analisar a possibilidade desta conversão mostrou que algumas configurações do micro-controlador que são feitas habilitando ou desabilitando determinados bits em registradores também podem ser feitas no MC9S08QE128, mas de modo diferente. Isso acarretaria em ter que alterar algumas definições da própria API, o que seria extremamente oneroso e fugiria completamente do foco deste projeto.

Ao procurar uma outra alternativa para a execução dos testes e uso das placas fabricadas, o autor encontrou a biblioteca SMAC. Esta biblioteca, criada e mantida pela empresa Freescale, é uma pilha de código escrita em linguagem C e é utilizada para funcionamento dos transceptores RF que utilizam a tecnologia IEEE 802.15.4, desta mesma empresa. Esta pilha de código tem por objetivo ser extremamente simples e fácil de ser utilizada, chegando ao ponto de podermos apenas escolher um canal de transmissão e enviar dados ao meio, sem precisar tomar nenhum cuidado de controle de ocupação de meio.

Em contrapartida, a biblioteca SMAC é disponibilizada portada para uma vasta gama de micro-controladores da empresa Freescale e entre eles está o utilizado nas novas placas. Com isso, surgiu uma nova alternativa para fazer os testes da rede, não tão prática, eficiente e elegante quanto se estivéssemos utilizando a API IEEE 802.15.4, mas que com os devidos cuidados pode ser tão completa quanto a mesma. Veremos no item a seguir como que a biblioteca SMAC foi empregada para o teste da rede.

## 6.1 Uso da biblioteca SMAC

O uso desta biblioteca para o teste da rede é extremamente demorado, por uma série de fatores que veremos na seqüência da análise. A implementação do código de teste utilizando a SMAC é razoavelmente simples e envolve a captura dos pacotes de associação a um coordenador, já que a mesma não implementa em seu pseudo-MLME um mecanismo de associação.

Essa captura foi uma das partes mais demoradas de todo o teste, pois até então, no projeto, o autor não tinha lidado diretamente com os pacotes que são transmitidos pelo meio e, utilizando a API IEEE 802.15.4, seria extremamente difícil de capturá-los. Para a captura do pacote, foi definido à aplicação SMAC o endereço *0xCAFE*, que é o endereço do coordenador PAN. Logo após, a aplicação SMAC foi colocada em modo de recepção, e fizemos um nodo transmitir o pacote de associação ao coordenador, de forma que pudéssemos capturar este pacote.

De posse da mensagem de associação, o que teria que ser feito a seguir era alterá-la nos seus bits de endereço para que solicitássemos a associação diretamente ao nodo que estava associado ao coordenador PAN.

A seqüência de passos adotada para efetuar os testes foi a seguinte:

1. Inicialização do coordenador PAN e de um outro nodo associado a ele;
2. Verificação do canal escolhido pelo coordenador PAN;
3. Inserção do canal selecionado no código da aplicação SMAC, recompilação e regravação do mesmo no módulo;
4. Envio do pacote de associação modificado ao nodo associado ao coordenador PAN sem fazer qualquer tipo de varredura na rede, já que sabemos de antemão o endereço dele.
5. Envio, a partir da interface “web”, de um comando à aplicação SMAC, que modifica o status do seu LED.

Durante estes passos, vale ressaltar que não há o aguardo da resposta de associação pela aplicação SMAC. O teste do envio da resposta de associação pelo coordenador PAN e pelo nodo associado a ele foi efetuado de maneiras distintas para os dois. No caso do coordenador PAN, juntamente com a resposta de associação, ele envia uma mensagem através da porta RS-232 indicando o endereço que está sendo atribuído ao novo nodo. No caso do nodo que está associado ao coordenador PAN, ao enviar a resposta de associação, ele pisca o seu LED por três segundos.

Além disso, para os testes, o endereço da aplicação SMAC foi fixado em *0x0002*, pois sabemos por construção que o primeiro endereço é atribuído ao primeiro nodo que se associa ao coordenador PAN e, portanto, a aplicação SMAC recebe o segundo endereço.

## 6.2 Resultados

Pelos testes realizados, todas as funcionalidades implementadas da rede foram verificadas e com seu funcionamento comprovado. Infelizmente, pela falta de recursos, os testes não puderam ser mais exaustivos. O ideal, e mais interessante, seria montar uma rede mais abrangente, com mais nodos.

### 6.2.1 Tempo de transmissão de dados de um nodo ao outro

O tempo de transmissão de uma mensagem de dados de um nodo ao outro foi medido iniciando um contador em hardware programando-o para realizar um incremento a cada cento e vinte e oito ciclos de relógio, que no caso é 16 MHz. No caso, o tempo foi medido da seguinte forma:

O coordenador envia ao nodo duas mensagens contendo cem bytes de dados cada. O total de duas mensagens é pensando que temos que esperar o período inteiro de “polling”, a fim de obtermos uma boa precisão estatística prevendo pouca variação no pior caso (até para obtermos uma melhor estimativa do custo de processamento). Logo após o recebimento da confirmação do envio da primeira mensagem, o contador é acionado, a segunda mensagem é enviada, e o contador é desligado quando recebemos a confirmação do envio da segunda mensagem. Uma importante informação a ser levada em consideração é que o nodo que recebe as mensagens está operando com o período constante de “pollings” de 820 ms.

A quantidade de dados escolhida foi de 100 bytes, pois o tamanho máximo definido pelo padrão para o “data payload” é de 102 bytes, e deseja-se chegar perto do pior caso.

O experimento foi repetido dez vezes e podemos conferir pela Tabela 6.1 abaixo o número de contagens que o contador efetuou:

Tabela 6.1: Tempos de transmissão de mensagens.

Número da amostra:	Resultado do contador (em base hexadecimal):	Resultado do contador:
1	19DFC	105980
2	1A0E2	106722
3	19FA4	106404
4	1A4FC	107772
5	1A3C7	107463
6	1989C	104604
7	196F5	104181
8	1A504	107780
9	19D29	105769
10	19B81	105345
<b>Média:</b>	-	<b>106202</b>
<b>Desvio-padrão:</b>	-	<b>1267.79</b>

Considerando que, em média, realizamos 106202 contagens, podemos estimar que o tempo de envio da mensagem considerando uma média do pior caso é de:

$$t = 106202 \times \frac{128}{16000000} = 849 \text{ ms}$$

Não se pode esquecer também que neste tempo está contido todo o tempo para se atravessar a pilha da API IEEE 802.15.4. Já que o nosso período lento de “polling” é de 820 ms, podemos concluir que obtivemos um “overhead” de aproximadamente 29 ms despendido para processamento.

Considerando que o pacote transmitido possui tamanho total de *131 bytes* (*100 bytes* de dados e *31 bytes* de “overhead”) e que o tempo médio de transmissão, conforme calculado acima, é de *849 ms*, então pode-se concluir que a taxa de transmissão, em *kbps*, é de:

$$t = \frac{131 \times 8}{0,849 \times 1000} = 1,235 \text{ kbps}$$

A taxa máxima teórica definida no padrão IEEE 802.15.4 é de *250 kbps*, portanto o mecanismo utilizado na rede criada neste projeto não ultrapassa o limite máximo deste padrão.

### 6.2.2 Tamanho de código

Em sistemas embarcados o tamanho de código é um fator importante a ser considerado. As novas placas, com seus *128 Kbytes* de memória FLASH, têm memória mais que suficiente para este projeto, porém as placas do laboratório LAIS, que possuem apenas *60 Kbytes* de memória FLASH precisaram ser levadas em consideração.

O tamanho total de memória ocupado pelos dois softwares é mostrado de acordo com a Tabela 6.2.

Tabela 6.2: Tamanho de código dos softwares dos módulos.

Software	Memória ocupada (bytes)
Coordenador PAN	56.500
Nodo comum	49.668

Uma das opções configuráveis da API IEEE 802.15.4 é o uso da segurança definida na norma. A inclusão de toda a biblioteca de segurança no binário do micro-controlador consome aproximadamente *7 Kbytes de memória*. Inicialmente, neste projeto, a segurança estava sendo utilizada, mas devido ao constante crescimento do tamanho do código, esta acabou, por fim, sendo removida.

### 6.2.3 Analisador espectral

Uma forma de comprovar que o coordenador PAN fez a escolha correta quanto à ocupação do canal de menor quantidade de energia, é utilizando um analisador de espectro. Este é um dispositivo que monitora toda a faixa de frequências do espectro de 2.4 GHz e é possível através dele verificar de forma visual os canais que estão em uso.

O analisador espectral utilizado foi o Wi-Spy, fabricado e comercializado pela empresa norte-americana MetaGeek. Este dispositivo funciona através da porta USB, e vem acompanhado de um software chamado Chanalyzer, também comercializado pela mesma empresa.

O Chanalyzer nos permite visualizar basicamente três diferentes gráficos (observados através da Figura 6.1) com diferentes informações a respeito do espectro. A descrição de cada um deles conforme a ordem que aparecem na Figura 6.1 é:

1. Visão Espectral. Contém um gráfico em estilo cascata que mostra amplitude em função do tempo para cada frequência.

2. Visão Topográfica. Contém um gráfico de amplitude em função da frequência similar ao da visão planar, mas ao invés de mostrar a amplitude atual de cada frequência, ele mostra a popularidade de cada coordenada de frequência/amplitude durante o tempo exibido.
3. Visão Planar. Contém um gráfico típico de amplitude em função da frequência. As linhas de cor amarela, verde e azul mostram respectivamente: amplitude atual, amplitude média e amplitude máxima.

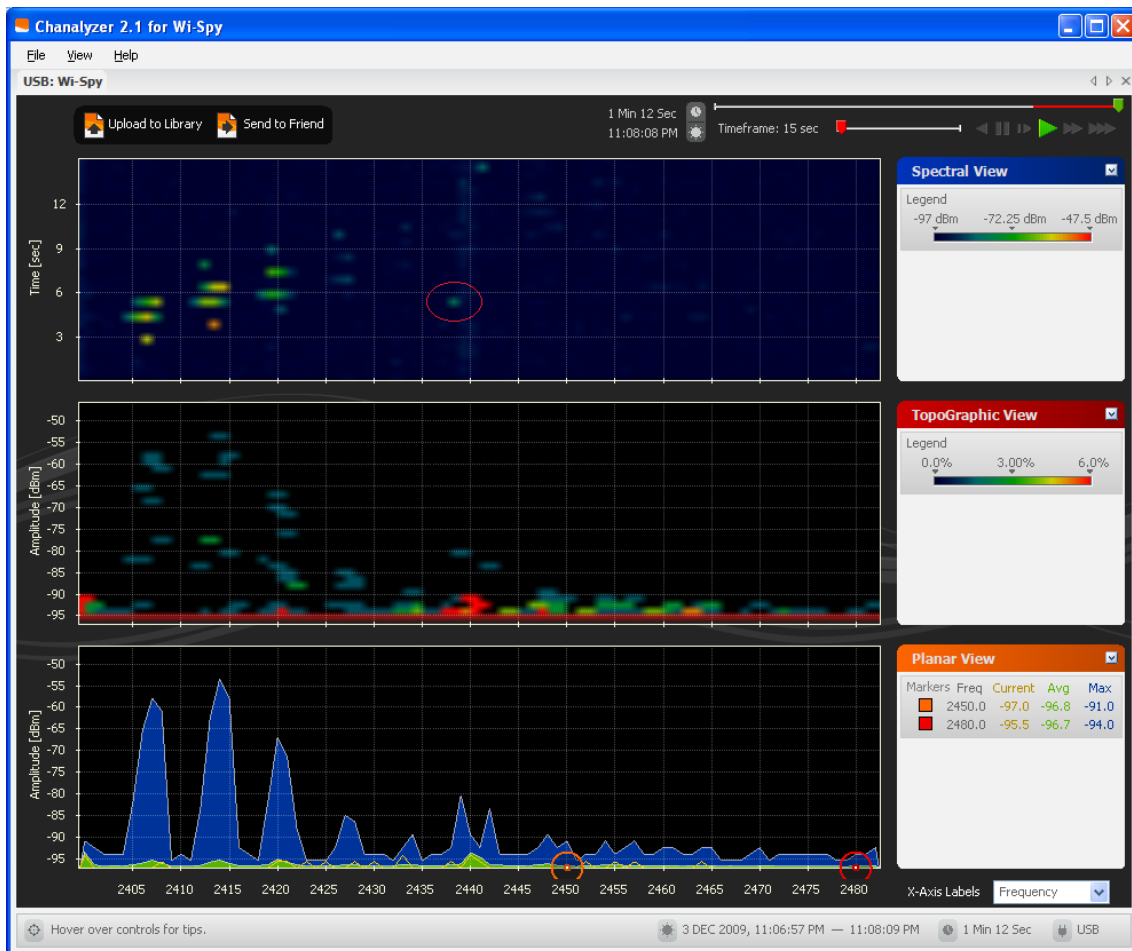


Figura 6.1: Analisador de espectro.

Como podemos observar através da Figura 6.1, no gráfico de visão espectral, a rede está ocupando o canal dez. Isso já era conhecido, pois esta informação está contida em uma das mensagens que o coordenador PAN transmite pela porta RS-232 logo após fazer a escolha do canal.

Podemos ver que o espectro não está muito ocupado, e que talvez a escolha tenha sido fácil porém, mesmo assim podemos comprovar a funcionalidade da varredura de energia no espectro.

O sinal de cor verde, demarcado pela elipse de cor vermelha, define a potência que é gasta no meio para realizar a transmissão. Neste caso, o sinal representa a transmissão do comando de acender o LED do exemplo implementado.

## 7 CONCLUSÃO

O objetivo deste projeto era criar a base de uma rede sem fio que cumprisse os requisitos de automação de um ambiente residencial. Este objetivo foi alcançado com sucesso, com as devidas simplificações que se mostraram necessárias para a conclusão do mesmo em tempo hábil.

O projeto não foi de fácil implementação, tendo o autor que enfrentar e solucionar diversos problemas, e contornar alguns outros. O projeto passou por muitas etapas diferentes. As principais, listadas na ordem em que foram abordadas, são:

- *Estudo do problema.* Analisar os quesitos de uma rede de automação residencial e pensar em como estender estes quesitos a redes sem fio.
- *Análise de tecnologias sem fio.* Estudo das diferentes novas tecnologias de transmissão de dados sem fio comercialmente disponíveis.
- *Análise do padrão IEEE 802.15.4.* Estudo das principais características do padrão de tecnologia sem fio escolhido.
- *Análise e escolha do hardware.* Estudo das opções de hardware, tomando como base os módulos do laboratório LAIS.
- *Especificação da rede.* Definição do comportamento geral da rede, além dos comportamentos específicos do coordenador PAN e dos nodos comuns.
- *Especificação do modo de controle da rede.* Definição da página “web” que se comunica com o software intermediário através de uma conexão TCP, que por sua vez se comunica com o coordenador PAN através da porta RS-232
- *Implementação do hardware.* Busca e compra dos componentes, confecção das placas de circuito impresso e solda.
- *Implementação da rede.* Codificação dos software do coordenador PAN, dos nodos comuns, da interface “web” e do software intermediário.
- *Especificação do método de testes.* Definição de como iria ser testada a rede, considerando o problema do porte da API IEEE 802.15.4 para as placas “design reference” da empresa Freescale.
- *Execução de testes.* Implementação e execução dos programas que testam o tempo de transmissão de mensagens de um nodo ao outro.
- *Obtenção de resultados.*

Em função de o projeto ser extenso, melhorias podem ser feitas a ele. Algumas foram pensadas durante a implementação do mesmo, porém devido ao custo de tempo que levariam, foram abandonadas e meramente registradas.



Entre as melhorias pensadas, estão:

- Construção de uma rede sinalizada;
- Utilização de um mecanismo de tempo de vida dos pacotes;
- Leiaute de hardware diferenciado;
- Hardware mais completo.

Entre essas quatro, a primeira já foi abordada na descrição do projeto, no item 5.2 da página 33. Por isso, serão comentadas melhor somente as três últimas.

Caso os nodos da rede percam a sua conexão enquanto seus respectivos coordenadores possuem mensagens para eles, em função de os coordenadores não receberem mais “pollings” destes nodos, as mensagens jamais serão enviadas. Isso pode fazer com que um dado nodo coordenador, acabe ficando sem memória. O mecanismo de tempo de vida de pacotes é interessante para solucionar este problema, pois podemos definir que um pacote fique armazenado no coordenador, somente durante um período de tempo pré-determinado.

Um dos grandes problemas encontrados no projeto foi a solda dos componentes SMD. Eles além de serem de difícil soldagem, também não permitem fácil localização de problemas no circuito. Através da alteração do leiaute do hardware, é possível a utilização de componentes maiores, que são de mais fácil manipulação.

A placa mais complexa do laboratório LAIS, conforme mencionado anteriormente, possui recursos que acabariam não sendo utilizados neste projeto. Em contrapartida, os demais módulos possuem poucos recursos, quando o ideal seria que possuíssem mais alguns. Uma outra melhoria prevista para o projeto, seria a de incluir nos módulos simples alguns recursos adicionais, como LEDs e mais pinos externos conectados diretamente às portas de entrada e saída do micro-controlador.

Finalmente, o projeto foi de extrema importância para o aprendizado do autor. Ele permitiu o aprofundamento em diversas áreas abordadas no curso de Engenharia de Computação. Além disso, com este aprendizado, diversas portas para uma melhor colocação profissional foram abertas.

## REFERÊNCIAS

- [1] PEREIRA, C. E. **Avaliação de Protocolos de Comunicação Wireless em Ambiente Industrial**. Fundação Empresa Escola de Engenharia. Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul. Porto Alegre, 2009.
- [2] INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERING. **Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)**: information technology – telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. New York, 2006.
- [3] HART COMMUNICATION FOUNDATION. **HART Field Communication Protocol Specification: TDMA Data Link Layer**. Austin, 2007.
- [4] LENNVALL T.; SVENNSON S.; HEKLAND, F. **A Comparison of WirelessHART and ZigBee for Industrial Applications**. ABB Corporate Research, Vasteras. Dresden, 2008.
- [5] 5. **The Benefits of HART Protocol Communication in 5t Instrumentation Systems**. Disponível em: <[http://www.5.com/PDFs/Catalogues/Hart\\_Tutorial.pdf](http://www.5.com/PDFs/Catalogues/Hart_Tutorial.pdf)>. Acessado em: jun. 2009.
- [6] TJENSVOLD, J. M. **Comparison of the IEEE 802.11, 802.15.1, 802.15.4 and 802.15.6 wireless standards**. Set. 2007. Disponível em: <<http://janmagnet.files.wordpress.com/2008/07/comparison-ieee-802-standards.pdf>>. Acessado em: jun. 2009.
- [7] ERGEN, S. C. **ZigBee/IEEE 802.15.4 Summary**. Set. 2004. Disponível em: <[www.sinemergen.com/zigbee.pdf](http://www.sinemergen.com/zigbee.pdf)>. Acessado em: jun. 2009.
- [8] WIRELESS SENSOR NETWORKS RESEARCH GROUP. **Wireless Sensor Networks Research Group**. Disponível em: <<http://www.sensor-networks.org>>. Acessado em: maio 2009.
- [9] ZIGBEE ALLIANCE. **ZigBee Specification**. Jan. 2008.
- [10] 10AYA, H.; TOMÁS, L.; SOUSA, J. P. **Sensorização sem Fios Sobre ZigBee e IEEE 802.15.4**. Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto. Portugal.
- [11] 11, J. **802.15.4 - An introduction**. Department of Computer Science - University of Copenhagen.
- [12] FREESCALE. **MC13202 2.4 GHz Low Power Transceiver for the IEEE® 802.15.4 Standard**: Freescale Semiconductor Technical Data. Dec. 2008.

[13] FREESCALE. **Simple Media Access Controller (SMAC):** SMAC User's Guide. Mar. 2008.

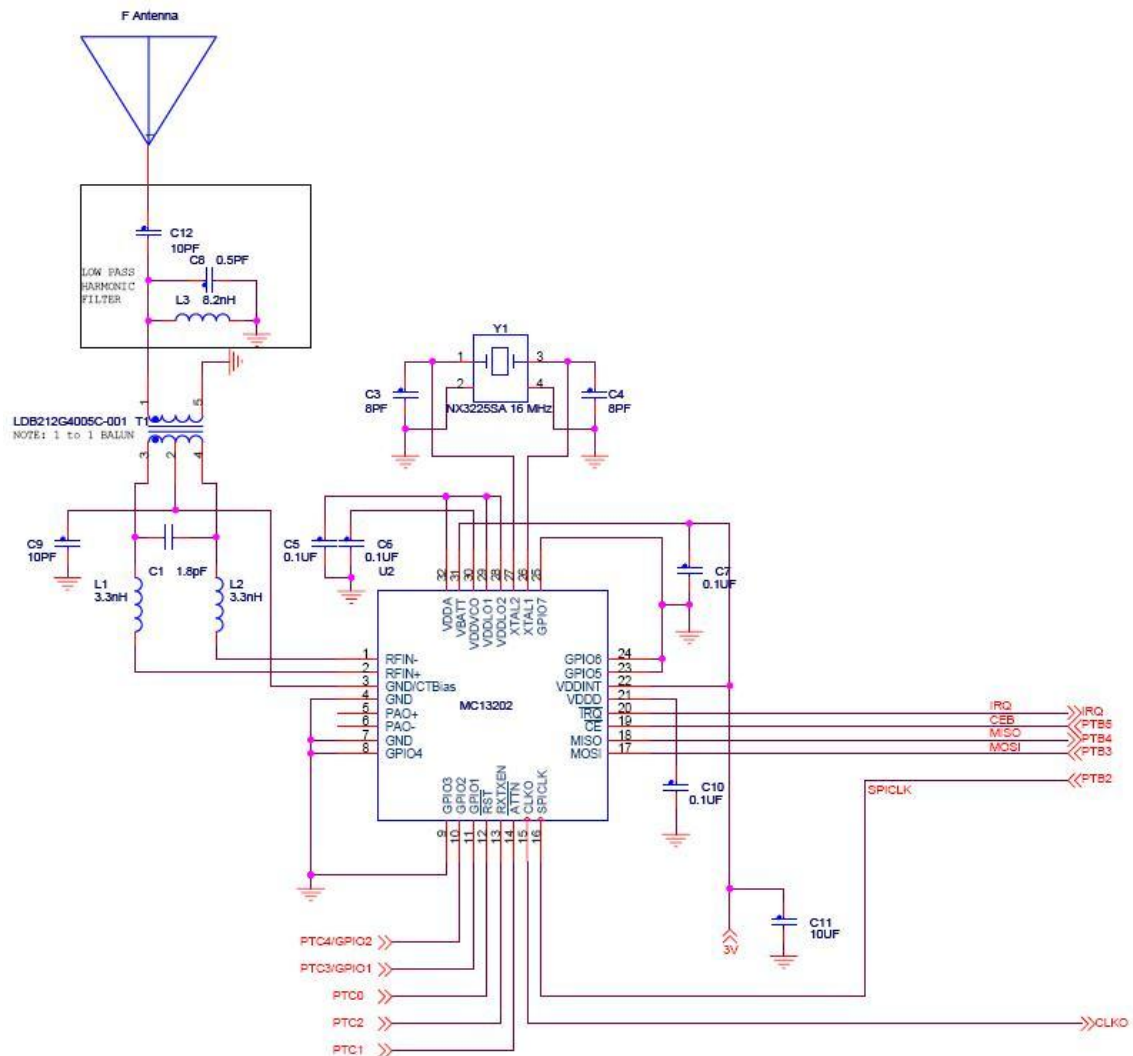
[14] FREESCALE. **802.15.4 MAC PHY Software:** Reference Manual. Set. 2005

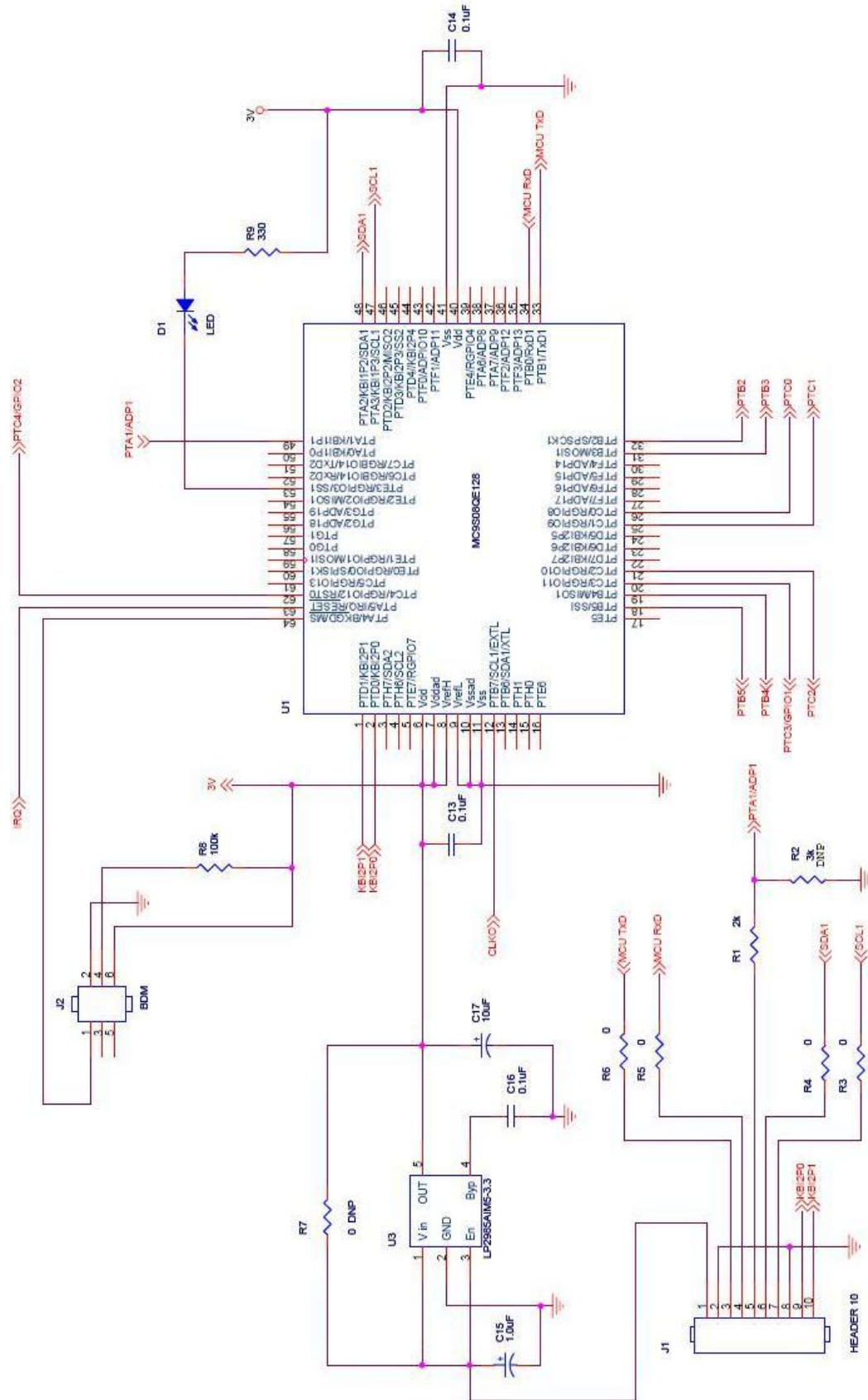
[15] FREESCALE. **Compact Integrated Antennas:** Designs and Applications for the MC1319x, MC1320x, and MC1321x. Jul. 2006.

[16] FREESCALE. **Open Source BDM-JM60.** Disponível em:  
<[http://www.freescale.com/files/microcontrollers/hardware\\_tools/community/OSBDM08.zip](http://www.freescale.com/files/microcontrollers/hardware_tools/community/OSBDM08.zip)>. Acessado em: outubro 2009.

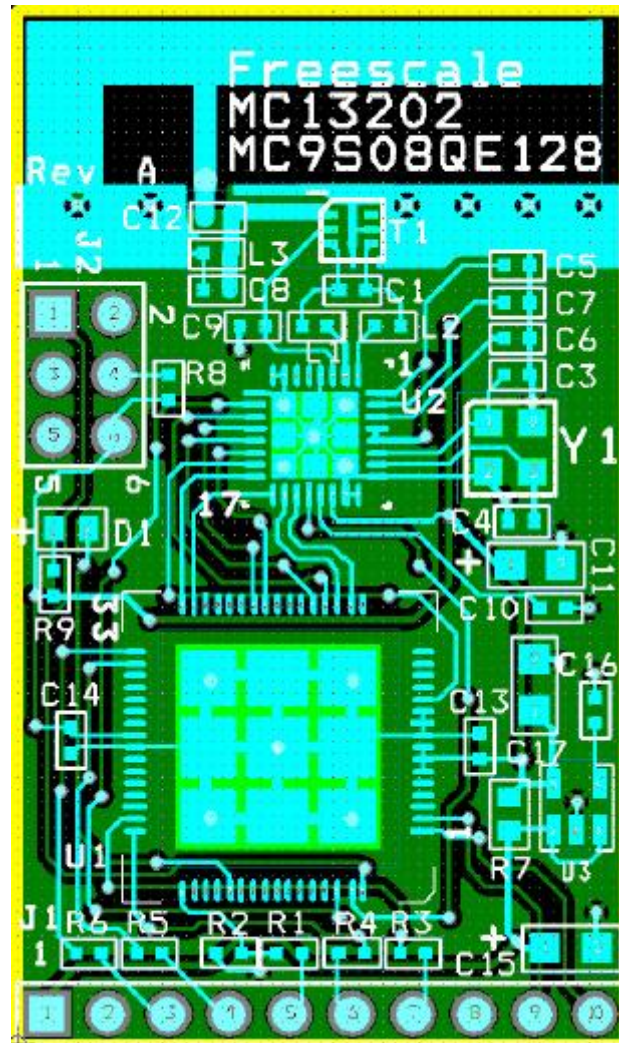
## ANEXO A – ESQUEMÁTICOS DOS MÓDULOS “DESIGN REFERENCE”

Abaixo seguem os esquemáticos das placas “design reference” da empresa Freescale. A primeira figura se refere ao esquemático da parte superior da placa enquanto que a segunda se refere à parte inferior.





A figura a seguir se refere ao leiaute do circuito da placas “design reference” da empresa Freescale.



## ANEXO B – ESQUEMÁTICO DO MÓDULO DO LABORATÓRIO LAIS

Abaixo segue o esquemático da placa do laboratório LAIS.

