

Este trabalho desenvolveu um sistema de aquisição de sinais biomédicos sem fio em uma rede de sensores utilizando comunicação *Zigbee* implementada em uma plataforma de aquisição e processamento de dados biomédicos, utilizando módulos sensores comunicando-se a um computador. Criou-se também, um *software* para gerenciar os dispositivos presentes na rede e visualizar os sinais adquiridos ao usuário. Este trabalho integrou sensores de oxigenação do sangue, batimentos cardíacos e temperatura corporal, os quais foram escolhidos por serem considerados sinais vitais de fácil acesso e os testes de desempenho foram realizados em três voluntários a fim de verificar a transmissão, recepção e visualização dos dados, bem como a distância de comunicação.

Orientador: Dr Pedro Bertemes Filho

Coorientador: Dr Antonio Heronaldo de Souza

Joinville, 2013

ANO
2013

FELIPE CUBAS SCHULZ | PROPOSTA DE UMA REDE SEM FIO PARA
MONITORAMENTO DE SINAIS BIOELÉTRICOS



UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC

CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT

CURSO DE ENGENHARIA ELÉTRICA

DISSERTAÇÃO DE MESTRADO

**PROPOSTA DE UMA REDE SEM FIO
PARA MONITORAMENTO DE
SINAIS BIOELÉTRICOS**

FELIPE CUBAS SCHULZ

JOINVILLE, 2013

FELIPE CUBAS SCHULZ

**PROPOSTA DE UMA REDE SEM FIO PARA
MONITORAMENTO DE SINAIS BIOELÉTRICOS**

Dissertação apresentada para a
obtenção do título de mestre em
engenharia elétrica da
Universidade do Estado de
Santa Catarina, Centro de
Ciências Tecnológicas – CCT.

Orientador: Pedro Bertemes
Filho, Dr. Eng.

**JOINVILLE / SC
2013**

FICHA CATALOGRÁFICA

F439s

Schulz, Felipe.

Proposta de uma rede sem fio para monitoramento de sinais bioelétricos / Felipe Cubas Schulz; orientador: Pedro Bertemes Filho. – Joinville, 2013.

145 f. : il ; 30 cm.

Incluem referências.

Dissertação (mestrado) – Universidade do Estado Santa Catarina, Centro de Ciências Tecnológicas, Mestrado em Engenharia Elétrica, Joinville, 2013.

1. Redes sem fio. 2. Engenharia biomédica. I. Bertemes Filho, Pedro.

CDD 620.16

FELIPE CUBAS SCHULZ
PROPOSTA DE UMA REDE ZIGBEE PARA TRANSMISSÃO
SEM FIOS DE SINAIS BIOMÉDICOS

Dissertação apresentada ao Curso de Mestrado Profissional em Engenharia Elétrica como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração "Automação de Sistemas".

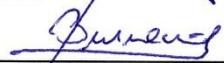
Banca Examinadora

Orientador:

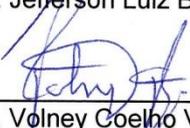


Prof. Dr. Antonio Heronaldo de Sousa
CCT/UDESC

Membros



Prof. Dr. Jefferson Luiz Brum Marques
UFSC



Prof. Dr. Volney Coelho Vincence
CCT/UDESC

Prof. Dr. André Bittencourt Leal - suplente
CCT/UDESC

Joinville,SC, 30 de agosto de 2013.

RESUMO

SCHULZ, Felipe Cubas. **“PROPOSTA DE UMA REDE SEM FIO PARA MONITORAMENTO DE SINAIS BIOELÉTRICOS”** 2013. 145 f. Dissertação (Mestrado em Engenharia de Elétrica – Área: Automação de Sistemas) – Universidade do Estado de Santa Catarina. Programa de Pós-graduação em Engenharia Elétrica, Joinville, 2013.

A automação de sistemas vem se disseminando muito nos últimos anos, estando presente em nosso dia a dia quando fazemos compras, vamos ao banco ou mesmo estando em nossas casas ou trabalho. Neste contexto vem crescendo o número de oportunidades de se inserir novas tecnologias e automação também na área da medicina, onde o monitoramento de pacientes torna diagnósticos mais fáceis, precisos e ergonômicos, principalmente quando utilizamos redes de transmissão de dados sem fios. Neste trabalho foi desenvolvido um sistema de aquisição de sinais biomédicos sem fio em uma rede de sensores utilizando comunicação *Zigbee*. Foi implementado uma plataforma de aquisição e processamento de dados biomédicos, utilizando módulos sensores de comunicação sem fio com um computador. Também, um *software* foi desenvolvido para gerenciar os dispositivos presentes na rede e visualizar os sinais adquiridos ao usuário. Este trabalho integrou sensores de oxigenação do sangue, batimentos cardíacos e temperatura corporal, os quais foram escolhidos por serem considerados sinais vitais de fácil acesso. Testes de desempenho da rede de sensores foram realizados a fim de verificar a transmissão, recepção e visualização dos dados, bem como a distância de comunicação. Também, aquisição de sinais foram realizados em 3 voluntários saudáveis com idades de 28, 25 e 65 anos e os resultados foram comparados com os sinais adquiridos por equipamentos comerciais. Os resultados obtidos mostraram que o sensor de oxigenação do sangue apresentou desempenho similar para os 3 voluntários quando comparados ao sistema comercial. O sensor de batimentos cardíacos apresentou maior variação entre os valores médios pelo sistema proposto. O sensor de temperatura corporal apresentou leituras com um erro sistêmico de aproximadamente 2%. A utilização do protocolo de comunicação *Zigbee* em uma rede de sensores biomédicos permitiu o monitoramento contínuo de pacientes com maior flexibilidade de uso quando comparado a sistemas convencionais com tecnologias com fios. O alcance da rede chegou a aproximadamente 13

metros em um ambiente com paredes, sem o uso de roteadores. Outros sinais podem ser facilmente adicionados ao sistema e monitorados pela rede de sensores.

Palavras-chave: **Rede sem fios. Zigbee. Aquisição de sinais. Sensores Biomédicos.**

ABSTRACT

SCHULZ, Felipe Cubas. **“PROPOSAL OF A WIRELESS NETWORK FOR MONITORING BIOELECTRICAL SIGNALS”** 2013. 145 p. Dissertation (Master degree on Electrical Engineering – Subject: System Automation) – Universidade do Estado de Santa Catarina. Postgraduate Program on Electrical Engineering, Joinville, 2013.

Recently, automation systems have been widely investigated. Nowadays, they are present in our lives when shopping, banking, working at home or office. Technology innovations have been increased and embedded into medical and biological equipments, where patients can be better monitored for treatment and diagnosis. These allow precise and ergonomic equipments be designed, especially when using wireless sensor networks. It is developed in this work a biomedical signal acquisition system by using a wireless sensor network and the *Zigbee* technology for communication. It was implemented a system for acquiring and processing biomedical data by using commercial sensor modules for wireless communication to a host computer. Also, it was developed a graphical interface in order to manage the sensors of the network and to display the acquired signals to the user. This work has integrated there types of sensors, such as blood oxygenation, heart rate and body temperature. The sensors were chosen due to their easy accessibility and by the fact these type of signals are the most monitored in medicine. Performance tests of sensors network were made to investigate the transmission, reception and data visualization, as well as the communication distance. Also, signal acquisitions were performed in 3 healthy volunteers aged 28, 25 and 65 and the results were compared with the signals acquired by commercial equipments. The results showed that the performance of the blood oxygenation sensor was similar for the three volunteers when compared to the commercial systems. On the other hand, the measured heartbeat by the proposed system showed a greater variation. The body temperature sensor showed reliable readings with a maximum error of approximately 2%. The communication distance of the network was approximately 13 meters in an environment with walls and without the use of routers. It can be concluded that the use of *Zigbee* sensor network for monitoring bioelectrical signals can be easily implemented and embedded to

medical equipments due to its great flexibility when compared to systems which use wired technologies.

Keywords: Wireless networks. Zigbee. Signal acquisition. Biomedical sensors.

LISTA DE FIGURAS

Figura 1: Comparação de redes sem fio.....	18
Figura 2: Principais aplicações das redes <i>Zigbee</i>	24
Figura 3: Exemplo de topologia da rede <i>Zigbee</i> com os três tipos de dispositivos de rede.....	26
Figura 4: Camadas do protocolo <i>Zigbee</i> (Pilha <i>Zigbee</i>).....	27
Figura 5: PDU (<i>Protocol Data Units</i>) da camada PHY.....	28
Figura 6: Bandas de frequência <i>Zigbee</i>	29
Figura 7: Modelo de referência da camada PHY.....	30
Figura 8: Modelo de referência da camada MAC.....	32
Figura 9: Estrutura das mensagens na camada MAC.....	33
Figura 10: Modelo de referência da camada de rede.....	36
Figura 11: Modelo de referência da camada de aplicação.....	37
Figura 12: Modelo cliente servidor para a conexão de <i>clusters</i>	40
Figura 13: <i>Kit</i> de desenvolvimento 13213-SRB.....	41
Figura 14: Diagrama de blocos do <i>kit</i> de desenvolvimento MC13213 SRB.....	42
Figura 15: Sistema de aquisição de dados proposto.....	44
Figura 16: Janela de apresentação do <i>Software</i> Beekit da Freescale para configuração dos dispositivos <i>Zigbee</i>	45
Figura 17: Estrutura do <i>Zigbee</i> Test Client.....	46
Figura 18: Fluxograma da inicialização do coordenador da rede.....	51
Figura 19: Fluxograma do <i>software</i> dos dispositivos finais (sensores) da rede.....	53
Figura 20: Interface do “ <i>Command Console</i> ” da ferramenta “ <i>Test Tool</i> ” da Freescale.....	57
Figura 21: Setup de teste de validação da comunicação <i>Zigbee</i>	58
Figura 22: Gráfico de medição do alcance da rede em ambiente externo.....	69
Figura 23: Gráfico de medição do alcance da rede em ambiente interno sem paredes.....	69
Figura 24: Gráfico de medição do alcance da rede em ambiente interno com paredes.....	70
Figura 25: Formato de mensagem do ZTC.....	72
Figura 26: Formato genérico de mensagem do ZTC.....	72
Figura 27: Diagrama de blocos do <i>software</i> de interface com o usuário.....	73
Figura 28: Tela inicial da interface com o usuário.....	74
Figura 29: Aviso inicial para configuração dos dispositivos.....	75
Figura 30: Fluxograma da tela inicial da interface com o usuário.....	76
Figura 31: Fluxograma para a procura de novos dispositivos da rede.....	78

Figura 32: Tabela com as propriedades dos dispositivos da rede.	80
Figura 33: Tela de acompanhamento das aquisições de cada dispositivo	81
Figura 34: Tela para salvar os dados adquiridos em arquivo texto.....	81
Figura 35: Formato do arquivo de dados salvo pelo <i>software</i>	82
Figura 36: Tabela de atributos com apenas o coordenador inicializado	83
Figura 37: Tabela de atributos com o coordenador e mais dois dispositivos.....	84
Figura 38: Dados adquiridos no teste de monitoramento do sensor de temperatura.....	85
Figura 39: Monitoramento da temperatura quando aplicado um distúrbio no sensor.....	86
Figura 40: Pasta e nome do arquivo salvo.....	87
Figura 41: Arquivo gerado no teste de gravação de dados.....	87
Figura 42: Absorção da hemoglobina VS comprimento de onda.....	93
Figura 43: Ilustração da medição da oximetria no dedo.....	93
Figura 44: Sinal típico de um oxímetro de pulso	95
Figura 45: <i>Kit</i> de desenvolvimento MED-SPO2 da Freescale	96
Figura 46: Formato do pacote de comunicação serial.	96
Figura 47: Fluxo da comunicação entre o <i>kit Zigbee</i> e o <i>Kit</i> do Oxímetro	98
Figura 48: Foto do <i>kit</i> de medição de oxigenação no sangue acoplado ao <i>kit Zigbee</i> , via comunicação serial.	99
Figura 49: Resultados mostrados pela interface gráfica no teste de medição do sensor de oximetria	100
Figura 50: Imagem do arquivo de dados gravado durante o teste de medição oxigenação do sangue.....	101
Figura 51: Arritmias cardíacas.	102
Figura 52: Princípio de funcionamento do sensor de batimentos cardíacos.....	103
Figura 53: Sensor utilizado para o monitoramento dos batimentos cardíacos.....	104
Figura 54: Sinal medido no sensor de batimentos cardíacos utilizando um osciloscópio.....	105
Figura 55: Foto do circuito de medição de batimentos cardíacos acoplado ao <i>kit Zigbee</i>	105
Figura 56: Resultados mostrados pela interface gráfica durante a aquisição de dados do teste de batimentos cardíacos.	106
Figura 57: Imagem do arquivo de dados gravado durante o teste de medição dos batimentos cardíacos.	107

Figura 58: Variação da resistência ôhmica em função da temperatura	110
Figura 59: Circuito de medição da temperatura corporal.....	110
Figura 60: Foto do dispositivo com o sistema de aquisição de temperatura	113
Figura 61: Resultados mostrados pela interface gráfica durante o teste do sensor de temperatura.....	114
Figura 62: Imagem do arquivo de dados gravado durante o teste do sensor de temperatura	115
Figura 63: Tela de identificação dos dispositivos do teste com os três sensores conectados a rede.....	116
Figura 64: Interface com o usuário e gráficos do teste do sistema com os três dispositivos conectados a rede.	117
Figura 65: Imagem do arquivo de dados gravado durante o teste com os três dispositivos sensores conectados a rede.....	118
Figura 66: Foto do sensor de temperatura comercial digital utilizado na medição da temperatura corporal.	119
Figura 67: Foto do sensor comercial utilizado para a medição da oxigenação no sangue e dos batimentos cardíacos. Modelo CMS50F.	119
Figura 68: Resultados mostrados pela interface gráfica durante o teste de medição da oxigenação do sangue nos três voluntários.....	120
Figura 69: Resultados mostrados pela interface gráfica durante o teste de medição da frequência cardíaca nos três voluntários.....	121
Figura 70: Resultados mostrados pela interface gráfica durante o teste de medição da temperatura corporal nos três voluntários.....	122

LISTA DE TABELAS

Tabela 1: Principais características da rede <i>Zigbee</i>	24
Tabela 2: Faixas de frequência do protocolo <i>Zigbee</i>	29
Tabela 3: Range de endereçamento <i>Zigbee</i>	34
Tabela 4: Perfil de aplicação <i>Zigbee</i> “ <i>Home Automation</i> ”	38
Tabela 5: Clusters disponíveis na aplicação “ <i>Home Automation</i> ”	39
Tabela 6: <i>Clusters</i> do dispositivo coordenador da rede	49
Tabela 7: <i>Clusters</i> do dispositivo sensor de temperatura corporal	50
Tabela 8: <i>Clusters</i> do dispositivo sensor de oxigenação no sangue	50
Tabela 9: <i>Clusters</i> do dispositivo sensor de batimentos cardíacos	50
Tabela 10: Descrição das etapas de inicialização do coordenador da rede.	52
Tabela 11: Descrição das etapas de inicialização dos dispositivos finais (sensores) da rede.	53
Tabela 12: Descrição das etapas do fluxograma da tela inicial da interface com o usuário	76
Tabela 13: Descrição das etapas do fluxograma para a procura de novos dispositivos da rede	79
Tabela 14: Diferentes sinais bioelétricos.	90
Tabela 15: Valores normais para a temperatura corporal.	108
Tabela 16: Curva do NTC e cálculo da precisão de leitura	112
Tabela 17: Média dos valores lidos no sensor de oxigenação do sangue nos testes com os três voluntários.	122
Tabela 18: Média dos valores lidos no sensor de batimentos cardíacos nos testes com os três voluntários.	123
Tabela 19: Média dos valores lidos no sensor de temperatura corporal nos testes com os três voluntários.	124
Tabela 20: Comandos do ZTC.....	135

LISTA DE ABREVIATURAS

APL - *Application Layer* – Camada de Aplicação
APS – *Application Sublayer* – Sub-Camada de Aplicação
APSDE – *Application Sublayer Data Entity* – Entidade de dados da Sub-Camada de Aplicação
APSME – *Application Sublayer Management Entity* – Entidade de gerenciamento da Sub-Camada de Aplicação
CSMA-CA - *Carrier Sense Multiple Access With Collision Avoidance* – Sensor de portadora de múltiplo acesso com evitador de colisão .
DSSS - *Direct Sequence Spread Spectrum* – Sequencia direta de espalhamento de espectro
ED – *Energy Detection* – Detecção de energia
FAPESC – Fundação de Apoio a Pesquisa Científica e Tecnológica do Estado de Santa Catarina
FFD - *Full Function Device* – Dispositivo de funções completas
IEEE - Instituto de Engenheiros Eletricistas e Eletrônicos
LAN - *Local Area Network* – Rede de area local
LQ – *Link Quality* – Qualidade da conexão
MAC - *Medium Access Control* – Camada de controle de acesso ao meio
MAN - *Metropolitan Area Network* – Rede de área metropolitana
NLME – *Network Layer Management Entity* – Entidade de gerenciamento da Camada de Rede
NTC - *Negative Temperature Coefficient*
NWK - *Network Layer* – Camada de Rede
PAN - *Personal Area Network* – Rede de área pessoal
PD - *PHY Data* – Dados PHY
PDU - *Protocol Data Unit* – Unidade do protocolo de dados
PHY - *Physical Layer* – Camada Física
PLME - *PHY Layer Management Entity* – Entidade de gestão da camada PHY
RF - *Radio Frequency* – Frequencia de Radio
RFD - *Reduced Function Device* – Dispositivo de funções reduzidas
SAMU - Serviço de Atendimento Móvel de Urgência
SAP – *Service Access Point* – Ponto de acesso ao serviço
SDS – Secretaria de Estado do Desenvolvimento Econômico e Sustentável
SSP - *Security Server Provider* – Camada de Segurança
TIC - Tecnologia da informação e comunicação

UART - Universal Asynchronous Receiver/Transmitter – Transmissor /
Receptor Assíncrono Universal

WAN - *Wide Area Network* – Rede de área alargada

ZDP – Zigbee Device Profile – Perfil de Dispositivo Zigbee

ZTC – Zigbee Test Client – Cliente de teste Zigbee

SUMÁRIO

CAPITULO 1 - INTRODUÇÃO	16
1.1 OBJETIVOS.....	20
1.1.1 Objetivo geral	20
1.1.2 Objetivos específicos.....	20
1.2 JUSTIFICATIVAS.....	21
1.3 ORGANIZAÇÃO DO TRABALHO	21
CAPITULO 2 – PROTOCOLO DE COMUNICAÇÃO SEM FIO ZIGBEE.....	23
2.1 PROTOCOLO DE COMUNICAÇÃO SEM FIOS ZIGBEE ...	23
2.1.1 Tipos de dispositivos.....	25
2.2 CAMADAS DO PROTOCOLO ZIGBEE.....	26
2.2.1 Camada PHY	27
2.2.2 Camada MAC	31
2.2.3 Camada de rede (NWK)	34
2.2.4 Camada de segurança	36
2.2.5 Camada de aplicação.....	37
2.3 KIT DE DESENVOLVIMENTO.....	40
2.4 IMPLEMENTAÇÃO DA REDE SEM FIO PARA AQUISIÇÃO DE SINAIS	43
2.4.1 Configuração dos dispositivos	45
2.4.2 INICIANDO A REDE.....	54
2.5 TESTES DE DESEMPENHO DA REDE.....	56
2.5.1 Teste de validação da comunicação dos dispositivos via protocolo Zigbee	56
2.5.2 Teste de medição do alcance da rede	67
CAPÍTULO 3 – SOFTWARE DE INTERFACE COM O USUÁRIO	71
3.1 COMUNICAÇÃO ENTRE O SOFTWARE DE INTERFACE COM USUÁRIO E O FIRMWARE ZIGBEE.....	71
3.2 FUNCIONAMENTO DO SOFTWARE DE INTERFACE COM O USUÁRIO.....	72
3.3 TESTES DO SOFTWARE DE INTERFACE COM O USUÁRIO.....	82
3.3.1 Teste de identificação dos dispositivos da rede	82
3.3.2 Teste de aquisição de dados dos dispositivos.....	84
3.3.3 Teste de geração de relatório de dados.....	86
CAPITULO 4 – SINAIS BIOELÉTRICOS.....	89
4.1 SISTEMA DE MONITORAMENTO DA OXIGENAÇÃO DO	

SANGUE.....	91
4.1.1 Circuito utilizado para a aquisição de dados de oxigenação do sangue.....	95
4.1.2 Teste de medição da oxigenação do sangue	99
4.2 SISTEMA DE MONITORAMENTO DOS BATIMENTOS CARDÍACOS.....	101
4.2.1 Circuito utilizado para aquisição de dados de batimentos cardíacos.....	103
4.2.2 Teste de medição batimentos cardíacos	106
4.3 SISTEMA DE MONITORAMENTO DA TEMPERATURA CORPORAL	107
4.3.1 Circuito utilizado para a aquisição da temperatura corporal	109
4.3.2 Teste de medição da temperatura corporal.....	113
4.4 TESTE DO SISTEMA COMPLETO	115
4.4.1 Testes do sistema com três voluntários	118
CAPITULO 5 – DISCUSSÕES E CONCLUSÕES	125
REFERENCIAS BIBLIOGRÁFICAS	128
APÊNDICE – LINK QUALITY INDICATOR (LQI).....	134
ANEXO – COMANDOS DO ZIGBEE TEST CLIENT (ZTC)	135

CAPITULO 1 - INTRODUÇÃO

A Tecnologia da Informação e Comunicação (TIC) atua transversalmente, dando suporte a todas as áreas do conhecimento. Com base neste entendimento, o Governo do Estado de Santa Catarina, através da SDS – Secretaria de Estado do Desenvolvimento Econômico e Sustentável e da Fapesc – Fundação de Apoio a Pesquisa Científica e Tecnológica do Estado de Santa Catarina – acreditam na importância da TIC como elemento chave para alavancar as outras áreas de inovação tecnológica, e promover o desenvolvimento do Estado. Desta forma, iniciou o fomento a um conjunto de projetos integrados e focados no tema “Tecnologia da informação e comunicação para serviços multiplataforma em larga escala: em direção a hipermídia ubíqua”, com aplicações de TICs de interesse público e voltados a pesquisa aplicada e desenvolvimento tecnológico, de natureza multidisciplinar, que tenham caráter estruturante para o Sistema de C,T&I no Estado de Santa Catarina.

Neste âmbito, um dos subprojetos propostos é o MONITORA – Monitoramento remoto e processamento de informações para computação. Este projeto tem como objetivo permitir o processamento de informações obtidas de pontos remotos (por exemplo: monitoramento de pacientes em domicílio (“*Home-care*”) ou em Serviço de Atendimento Móvel de Urgência – SAMU; sistema de gerenciamento de equipamentos e tecnologias médico-hospitalar, etc.). Para este fim técnicas para redes de sensores, comunicação para sistemas sem fio; telefonia móvel e ferramentas para realidade virtual operando em portais interativos para disponibilizar informações a *internet* ou outras plataformas como TVDi devem ser estudadas.

Alta mobilidade na obtenção, processamento e apresentação de dados e informações são características de suma importância nestes tipos de sistemas. Exemplos importantes de plataformas com tais requisitos residem na área da saúde pública. Os sistemas de atendimento domiciliar de saúde e os serviços móveis de atendimento de urgências médicas são entendidos pelo sistema público de saúde como essenciais em garantir a qualidade, a universalidade dos serviços e a qualidade de vida da população. Estes serviços dependem fortemente de informações originadas a partir de diferentes tecnologias, originadas em larga escala a partir de numerosas fontes, e que devem ser apresentadas de modo rápido e adequado a um grande número de agentes.

Devido a mudança do perfil epidemiológico, provocada pelo

aumento de expectativa de vida e consequente envelhecimento da população e cronificação de certas doenças, o monitoramento contínuo dos pacientes e as modalidades alternativas a hospitalização que utilizam o domicílio, como lugar de cuidado a saúde, tem apresentado rápido e expressivo crescimento no Brasil.

As tecnologias sem fios estão para transformar a forma como cuidamos da saúde (SMITH, 2011). Diversos dispositivos sensores serão incluídos no nosso dia a dia e permitirão por exemplo, o monitoramento dos fluidos no pulmão de um paciente cardíaco, permitindo assim que o médico seja alertado e altere a dosagem da medicação necessária para o paciente naquele momento. Da mesma forma, sensores colocados na sua residência podem confirmar que o seu corpo se adaptou bem a alteração da medicação, ou que você pode estar próximo de se tornar um diabético. O *check up* geral que normalmente é feito uma vez por ano para verificar a saúde do paciente, tende a se tornar parte do cotidiano das pessoas.

Este tipo de monitoramento torna a medicina mais assertiva, pois possibilita que o paciente seja monitorado regularmente, ajudando na prevenção de doenças e na melhora da qualidade de vida da população, proporcionando ações preventivas e não mais corretivas.

O processamento de fluxo contínuo de dados está em expansão e está voltado para o processamento de informações produzidas por dispositivos que geram grandes volumes de dados em alta velocidade e com tempo de vida útil limitado (LEITE, 2010). Por exemplo, as informações geradas por sensores que coletam os sinais vitais de pacientes são sequencias contínuas e ilimitadas de dados.

Tradicionalmente, tais informações requerem dispositivos, equipamentos, *softwares* e aplicações especiais para monitorá-las, que processam e reagem a entrada contínua de diversas origens e que sejam capazes de propagar e compartilhar essas informações por diferentes meios a fim de permitir amplo acesso aos dados adquiridos.

Neste sentido se faz necessário o uso de sistemas de transmissão robustos e que facilitem o processo de automação de coleta dados. Em se falando de automação, existem conceitos que são fundamentais, tais como: aquisição de dados para controle de processos; monitoramento e processamento de sinais; redução de custos; otimização de processos (VALENTIN, 2009). Neste contexto sistemas de transmissão sem fios vem se tornando acessíveis e estão disponíveis em diversos dispositivos em nosso dia a dia.

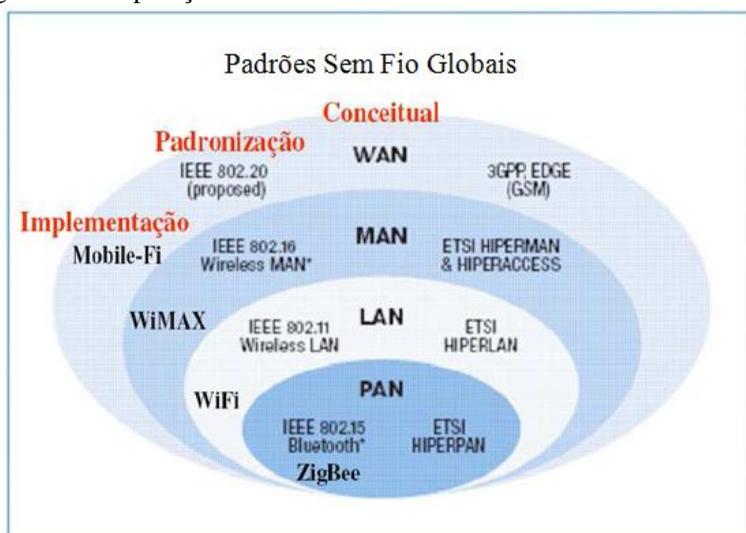
O monitoramento remoto, também chamado de telemetria pode

ser aplicado utilizando-se diversos métodos de transmissão. Podem ser citadas: transmissões acústicas (ISTEPANIAN *et al.*, 1997), luz infravermelha (PARK *et al.*, 1998) e radiofrequência. A mais utilizada é a radiofrequência, principalmente por ser menos susceptível à presença de obstáculos entre transmissor e receptor (MARTINCOSKI, 2003).

Dentro da radiofrequência existem diversas tecnologias disponíveis para efetuar a transmissão de dados, variando em área de abrangência, capacidade de transmissão de dados, custo e facilidade de implementação. A escolha da tecnologia que melhor satisfará a aplicação depende dos requisitos a serem atendidos pelo projeto e da experiência do projetista.

A Figura 1 mostra a comparação de redes sem fio. Nela pode-se verificar diferentes tipos de tecnologias e suas respectivas classificações. De forma resumida, a “Wide Area Network” (WAN), ou Rede de Longa Distância, também conhecida como Rede Geograficamente Distribuída, é uma rede de computadores que abrange uma grande área geográfica, com frequência abrangendo um país ou continente, variando de 1 km a 40.000 km e com taxas de transferências consideradas moderadas ou baixas (PUTTINI,2000).

Figura 1: Comparação de redes sem fio.



Fonte: BRANQUINHO, et al , 2005

A *Metropolitan Area Network* (MAN), ou Rede de Área

Metropolitana é o nome dado às redes que ocupam o perímetro de uma cidade. São mais rápidas e permitem que empresas com filiais em bairros diferentes se conectem entre si, normalmente com atuação de 1 km a 50 km.

A *Local Area Network* (LAN), ou Rede de Área Local cobre uma área geográfica reduzida, tipicamente um escritório ou uma empresa, e interligam um número não muito elevado de entidades. São usualmente redes de domínio privado, normalmente tendo abrangência de 10 m a 10 km.

A *Personal Area Network* (PAN), ou Rede Pessoal é uma tecnologia de rede formada por nós (dispositivos conectados à rede) muito próximos uns dos outros (geralmente não mais de uma dezena de metros). Por exemplo, um computador portátil conectando-se a e este a uma impressora. Estas redes geralmente estão organizadas ao redor de uma pessoa e tipicamente envolvem um computador móvel, um celular ou outro dispositivo computacional de mão.

Segundo MARTINCOSKI (2003), os sistemas de telemetria médica por radiofrequência surgiram na década de 70, fazendo uso de dispositivos de transmissão sobre uma única de rádio com modulação analógica. O sucesso de um bom sistema de informação e comunicação é conseguir conciliar diferentes tecnologias de aquisição e transmissão de dados proporcionando aos seus usuários facilidade no acesso a informação.

Neste contexto a tecnologia *Zigbee*, que é uma tecnologia que se enquadra dentre as redes PAN, surgiu em 2004, como um novo padrão definido pelo IEEE802.15.4 para transmissão digital de curto alcance, desenvolvido por um grupo de trabalho chamado *Zigbee Alliance*, composto por vários fabricantes renomados de dispositivos eletrônicos. Esta tecnologia incorpora diversas vantagens frente às demais, principalmente quanto ao seu baixíssimo consumo de energia, na ordem de micro amperes e baixas taxas de transmissão, na ordem de 50 kbps (ANDRIGHETTO, 2008). Isso faz do *Zigbee* uma atraente alternativa para projetos de monitoração e sensoriamento remoto, em que os requisitos passam a ser superdimensionados com o emprego das demais tecnologias, como por exemplo, *Bluetooth* e *WiFi*.

Dentre as principais aplicações *Zigbee*, cita-se: a automação e controle predial; controle industrial; periféricos para computadores; controles remotos; e saúde pessoal.

1.1 OBJETIVOS

1.1.1 Objetivo geral

O objetivo deste trabalho é desenvolver um sistema de aquisição de sinais bioelétricos sem fio em uma rede de sensores utilizando comunicação *Zigbee*. Desta forma será desenvolvida uma plataforma de aquisição e processamento de sinais bioelétricos, com o auxílio de módulos de sensores que se comunicam através de tecnologia sem fio a um computador central onde se terá um *software* que gerencia os dispositivos presentes na rede e mostra os sinais adquiridos ao usuário.

Estão englobados neste trabalho soluções de *hardware*, *software* necessárias para permitir a aquisição, transmissão e processamento de sinais bioelétricos de oxigenação do sangue, batimentos cardíacos e temperatura corporal, além do controle e gerenciamento da rede sem fio.

1.1.2 Objetivos específicos

De forma resumida os objetivos específicos do projeto são:

- Estudar o protocolo de comunicação *Zigbee*;
- Desenvolver uma rede sem fios para a transmissão de sinais bioelétricos;
- Criar uma interface em Labview para controle e gerenciamento dos dispositivos na rede *Zigbee*, além de gerenciar e manipular os sinais bioelétricos adquiridos;
- Integrar um sensor para medir a oxigenação do sangue (oxímetro), um sensor para medir a frequência cardíaca e um sensor para medir a temperatura corporal em uma rede de sensores sem fios utilizando o protocolo *Zigbee*.
- Desenvolver uma plataforma gráfica em Labview para recepção, processamento e visualização dos sinais e dos dispositivos da rede.

1.2 JUSTIFICATIVAS

O desenvolvimento de novas tecnologias de sensoriamento de sinais bioelétricos, proporcionadas pelo avanço da engenharia biomédica no mundo, aliado a grande facilidade de acesso a informações proporcionada pela difusão de dispositivos com grande capacidade computacional e interatividade, proporcionam uma excelente oportunidade para o desenvolvimento de sistemas portáteis capazes de fazer o monitoramento constante de grandezas até então monitoradas apenas de forma ocasional.

Aliado a isso, cresce rapidamente o número de pessoas idosas pelo mundo, com projeções apontando que a população mundial de pessoas acima de 65 anos alcançará 761 milhões de pessoas em 2025 (JUNG, 2008) e também a ansiedade das novas gerações por maior acesso a informações relacionados a sua saúde, com o objetivo de poder cuidar melhor de seus hábitos de saúde.

Neste contexto, o desenvolvimento de sistemas de monitoramento que possam ser inseridos no dia a dia das pessoas, dentro de suas casas, de forma não intrusiva ou desgastante se tornam necessários. Com isso, abre-se um novo mercado capaz de tornar a prevenção e a detecção precoce de doenças possível, tornando os tratamentos mais rápidos e eficientes. Torna-se possível também, a otimização do uso dos médicos, diminuindo o tempo de internação em hospitais e promovendo uma reeducação no âmbito da saúde reduzindo inclusive os custos de assistência médica.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em cinco capítulos, sendo que o capítulo 1 é dedicado a introdução ao tema do trabalho, descrevendo o que motivou o seu desenvolvimento, além de deixar claro seus objetivos gerais e específicos e suas justificativas, além de tratar da descrição de como o trabalho está organizado e estruturado.

O capítulo 2 descreve o protocolo de comunicação sem fios *Zigbee*, mostrando os padrões de mensagens utilizados e suas principais funções, além de explicar as diversas camadas da pilha *Zigbee*. Também será ressaltado a forma escolhida para a implementação do protocolo *Zigbee* na rede desenvolvida neste trabalho, assim como a descrição do *kit* de desenvolvimento utilizado e também alguns resultados de desempenho desta rede.

Já o capítulo 3 trata do desenvolvimento da interface com o usuário que, por sua vez, flexibilizará o uso dos recursos dos dispositivos da rede, além da visualização e interpretação dos sinais medidos, mostrando as diferentes telas de acesso e descrevendo as funcionalidades implementadas.

No capítulo 4 será descrito os sensores biomédicos utilizados neste trabalho. Neste capítulo é descrito o fenômeno físico e a técnica utilizada para a detecção da oxigenação do sangue, dos batimentos cardíacos e da temperatura corpórea, além de descrever também os circuitos utilizados e os resultados obtidos na medição destes sinais.

No capítulo 5 são expostas as discussões sobre os resultados obtidos neste trabalho, além das conclusões sobre o projeto desenvolvido e das oportunidades detectadas para trabalhos futuros.

CAPITULO 2 – PROTOCOLO DE COMUNICAÇÃO SEM FIO ZIGBEE

Este capítulo descreve o protocolo de comunicação sem fio *Zigbee*, mostrando os padrões de mensagens utilizados e suas principais funções, além de explicar as diversas camadas da pilha *Zigbee*. Também será ressaltado a forma escolhida para a implementação do protocolo *Zigbee* na rede desenvolvida neste trabalho, assim como a descrição do *kit* de desenvolvimento utilizado e também alguns resultados de desempenho da rede.

2.1 PROTOCOLO DE COMUNICAÇÃO SEM FIOS ZIGBEE

O protocolo de comunicação sem fios *Zigbee* foi concebido em 2004, fruto de um trabalho conjunto entre o IEEE e uma organização chamada *Zigbee Alliance*, a qual é formada por diversas empresas de âmbito mundial que uniram forças para escrever e padronizar um protocolo de comunicação sem fios com características específicas para um mercado de dispositivos sensores de baixo custo e baixa potência que pode ser utilizado em quase todos os mercados.

Segundo Norris (2005), *Zigbee* é um padrão para redes de telemetria sem fio, otimizado para baixo consumo de potência e um longo período de operação da bateria. A pilha protocolar *Zigbee* tem suporte a rede auto-organizável de dispositivos nas topologias de árvore, malha e estrela, permitindo uma instalação rápida de um sistema de telemetria sem fio interno.

Já Monsignore (2007), descreve o *Zigbee* como um padrão de comunicação *wireless* que provê uma rede de curto alcance e boa relação custo benefício. Foi desenvolvido com ênfase em aplicações de baixo custo alimentadas por bateria, tais como automação predial, controle industrial e comercial, assistência médica pessoal e sistemas de *tag* avançados.

A Figura 2 mostra as aplicações típica das redes *Zigbee*, as quais estão difundidas em áreas de automação, eletrônica de consumo, cuidados médicos pessoais, PCs e periféricos, controle industrial e controle residencial e comercial, podendo estar contidos em diversos dispositivos, permitindo controle e monitoramento de aplicações onde não sejam necessários altas taxas de transmissão.

Figura 2: Principais aplicações das redes *Zigbee*.

Fonte: Adaptado de VASQUES, 2010.

Algumas das principais características do protocolo *Zigbee* são mostradas de forma quantitativas na Tabela 1. Através desta referência pode-se verificar que esta tecnologia tem como diferencial em relação a outras tecnologias de comunicação sem fio de seu grupo de atuação (PAN) o alcance, que pode chegar a 100 metros de distância, o número de dispositivos na rede que pode chegar a 65.000 e ao perfil de alimentação, cujo consumo de energia permite que uma bateria tenha durabilidade de anos.

Tabela 1: Principais características da rede *Zigbee*

Característica	<i>Zigbee</i>
Alcance	10 a 100 metros
Taxa de Transmissão	20 a 250 kbps
Latência Típica	30 ms
Perfil de alimentação	Anos
Segurança	128 bits AES* e definível na camada de aplicação do usuário
Frequência de operação	868 MHz, 902-928 MHz ou 2,4 GHz ISM**
Complexidade	Simplex
Topologias de rede	Adhoc, estrela e malha híbrida
Número de dispositivos por rede	2 a 65.000
Escalabilidade /	Muito alta / sim

Extendabilidade	
Flexibilidade	Muito alta
Mobilidade e confiabilidade	Muito alta

*AES = Advanced Encryption Standard

**ISM = Industrial, Scientific and Medical

Fonte: Adaptado de Baker, 2005.

2.1.1 Tipos de dispositivos

O protocolo *Zigbee* trabalha com dois diferentes tipos de dispositivos, FFD (*Full Function Device*) ou RFD (*Reduced Function Device*), que por sua vez podem assumir três diferentes funções lógicas na rede, coordenador, roteador e *end device*, os quais diferem pelas atribuições na rede e complexidade de implementação. Toda rede é sempre formada por um único coordenador e pode conter vários roteadores e dispositivos finais (*end devices*).

O dispositivo coordenador é o dono da rede e é responsável pelo início, conexão de outros dispositivos, alocação de endereços e salvamento de mensagens até que possam ser enviadas. Este dispositivo é do tipo FFD, ou seja, é um dispositivo completo com todas as funcionalidades disponíveis na rede. Quanto ao consumo de energia, este dispositivo é normalmente alimentado pela rede de alimentação, visto que deve estar ativo em grande parte do tempo e deve garantir a confiabilidade da rede.

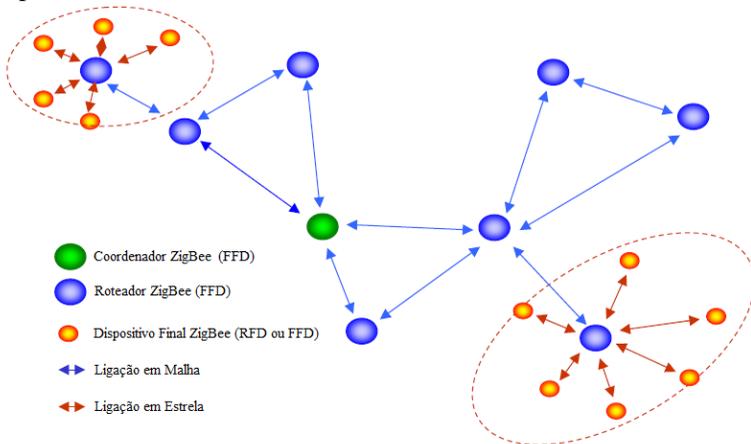
Os dispositivos roteadores são responsáveis por encaminhar mensagens entre nós da rede, ou por realizarem funções semelhantes aos dispositivos finais da rede, podendo atuar como um sensor ou atuador. Este tipo de dispositivo não tem capacidade de deter ou iniciar uma rede, mas sim procura uma rede para ser juntar. Quanto à funcionalidade é também do tipo FFD, apresentando grande parte das funcionalidades do protocolo, podendo se comunicar com diversos dispositivos presentes na rede. Para garantir a confiabilidade do envio de mensagens entre nós da rede este tipo de dispositivo também deve ser alimentado via rede de alimentação.

Os dispositivos finais da rede, chamados de *end devices*, são dispositivos que são colocados na rede para cumprir uma função específica, comumente sendo um dispositivo sensor ou que atua como atuador em algum tipo de carga ou controle. Este tipo de dispositivo é capaz de se comunicar com apenas um dispositivo, além de não ser

capaz de deter ou iniciar uma rede e não encaminhar mensagens. Para este tipo de dispositivo é possível utilizar a configuração chamada RFD, onde o requisito de memória do sistema é menor. É normalmente encontrado sendo alimentado por baterias.

A Figura 3 mostra a topologia de uma rede *Zigbee*, e nela pode-se verificar a presença dos três tipos de dispositivos descritos acima. Nota-se que em cada rede existe apenas um coordenador, e que os dispositivos finais são conectados a apenas um nó, o qual normalmente é um coordenador ou roteador, que permite o envio da mensagem deste dispositivo para outros dispositivos da rede.

Figura 3: Exemplo de topologia da rede *Zigbee* com os três tipos de dispositivos de rede.



Fonte: Adaptado de Vasques, 2010.

2.2 CAMADAS DO PROTOCOLO ZIGBEE

O protocolo de comunicação sem fio *Zigbee* foi desenvolvido em camadas (pilha), segundo o modelo de referência *Open Systems Interconnection* (OSI) de sete camadas, mas define apenas as camadas onde se obtém as funcionalidades da rede. Conforme mostra a Figura 4, o IEEE, através da diretiva 802.15.4 (IEEE Std802.15.4, 2003) padronizou as camadas da base da pilha, que são a Camadas Física – *Physical Layer* (PHY) e Camada de Enlace, também denominada de Camada de Controle de Acesso ao Meio – *Medium Access Control*

(MAC). A *Zigbee Alliance* desenvolveu as camadas intermediárias do protocolo, que são a Camada de Rede – *Network Layer* (NWK), Camada de Segurança – *Security Server Provider* (SSP) e Camada de Suporte a Aplicação – *Application Layer* (APL). Por fim fica a critério do usuário da rede definir a camada superior do protocolo que é a camada de aplicação.

Figura 4: Camadas do protocolo *Zigbee* (Pilha *Zigbee*)

Usuário	Camada de Aplicação
Zigbee Alliance	Camada de Suporte a Aplicação
	Camada de Rede (NWK) / Camada de Segurança (SSP)
IEEE 802.15.4	Camada MAC
	Camada PHY

Fonte: Adaptado de Vasques, 2010.

Cada camada da pilha realiza um conjunto de serviços para a camada superior. Estes serviços são classificados em dois tipos básicos: Entidades de Dados e Entidades de Serviços.

O entendimento do papel e do comportamento de cada camada da rede é importante para o projetista ter conhecimento de como a rede funciona e qual a melhor forma de modificar o protocolo para a implementação da sua aplicação.

2.2.1 Camada PHY

Segundo Costa Filho (2011), a camada PHY, ou camada física, é responsável pela comunicação propriamente dita entre os vários dispositivos contidos na rede. As seguintes tarefas são executadas nessa camada:

- Ativação/desativação do transceptor de rádio;
- Seleção da frequência de canal;
- Transmissão e recepção de dados;
- Detecção de energia (ED – *Energy Detection*) dentro do canal corrente;
- Indicador de qualidade do *link* (LQI – *Link Quality*)

- Avaliação de canal livre para CSMA-CA (*Carrier Sense Multiple Access With Collision Avoidance*).

Esta camada é especificada pelo IEEE 802.15.4 é então responsável pela transmissão das PDUs (*Protocol Data Units*), que são unidades de dados, transmitidas através de ondas de rádio, com o formato apresentado na Figura 5.

Figura 5: PDU (*Protocol Data Units*) da camada PHY

Bytes	4	1	1	n
Camada Física	Sequencia de Preâmbulo	Delimitador de início de quadro	Comprimento do quadro	MAC PDU
	sincronismo		informação	dados
	6+n			
	PHY PDU			

Fonte: Adaptado de Vasques, 2010.

O formato da PDU após a manipulação da camada física é, basicamente, um sinal de sincronismo, composto pela sequencia de preâmbulo (4 *bytes*) e pelo delimitador de início de quadro (1 *byte*), um bloco de informação (1 *byte*), e um bloco de dados ou *payload* de tamanho *n bytes* que representa a PDU vinda da camada superior (camada de enlace).

A camada PHY utiliza modulação DSSS (*Direct Sequence Spread Spectrum*) que incorpora em cada bit de dado um padrão de redundância e os espalha pela largura de banda utilizada. Esta redundância permite não só que o dado seja identificado como pertencente a um determinado nó, mas como, facilita a detecção de erros e permite que os equipamentos sejam muito simples, possibilitando implementações mais baratas.

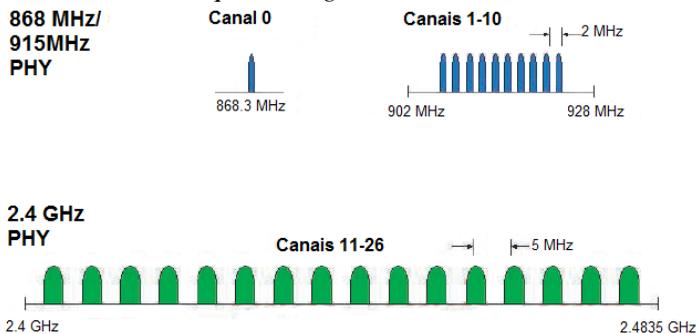
O IEEE definiu três bandas de frequências que podem ser utilizadas para as transmissões via *Zigbee*, conforme mostrado na Tabela 2. Estas faixas foram definidas de forma que a banda de 868 MHz deve ser utilizada na Europa, a banda de 915 MHz para ser utilizada na América do Norte, e a banda de 2,4 GHz que tem o uso permitido mundialmente.

Tabela 2: Faixas de frequência do protocolo *Zigbee*

PHY (MHz)	Banda de Frequência (MHz)	Canais	Modulação	Taxa de Transmissão (kbps)
868	868-868,6	1	BPKS	20
915	902-928	10	BPKS	40
2450	2400-2483,5	16	Q-qpsk	250

Fonte: Adaptado de COSTA FILHO, 2011.

As três bandas se diferenciam ainda pela taxa de transmissão, onde pode ser observado que a faixa de 2,4 GHz tem a maior taxa de transmissão alcançando 250 kbps e pelo número de canais disponíveis, onde pode-se chegar a até 16 canais na referida banda de 2,4 GHz, conforme também ilustra a Figura 6.

Figura 6: Bandas de frequência *Zigbee*

Fonte: Adaptado de Vasques, 2010.

Em operação, o protocolo *Zigbee* espalha os dados em todas as frequências da banda, sendo assim, o sinal resultante se assemelha a um ruído, tornando-se mais robusto a interferências. Após ser feito a DSSS, o sinal é modulado em uma portadora para transmissão.

Para indicar a qualidade de conexão, ao receber um pacote, o nó receptor envia de volta um pacote LQ (*Link Quality*) que determina a qualidade do sinal de conexão. O valor desse pacote é calculado pelas camadas superiores de Rede e Aplicações de acordo com a relação sinal-ruído e o valor do pacote ED (*Energy Detection*).

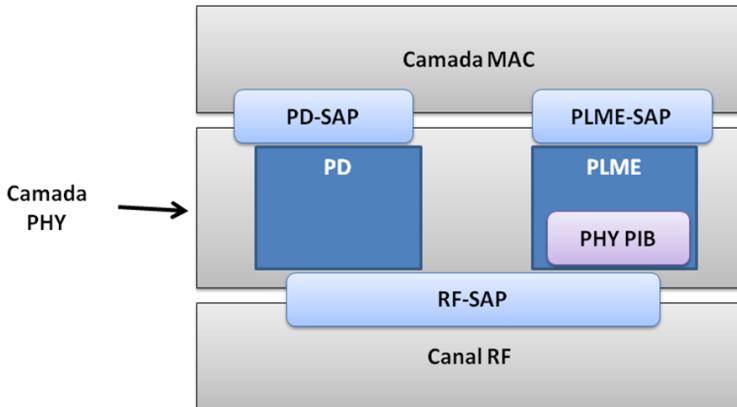
O valor da potência do canais é detectado e manipulado pela camada de rede, onde torna possível a escolha do canal a ser utilizado. O ED representa em 8 bits a relação em dB da potência recebida nos canais de acordo com os limites mínimos e máximos de sensibilidade. Dessa forma, os nós direcionam a informação do nível de qualidade do caminho dentro da topologia.

Para a avaliação de canais livres a camada física determina se os canais estão ocupados de acordo com o *Carrier Sense* dos sinais em DSSS ou caso o parâmetro ED esteja acima do limite do canal.

Existem três entidades associadas a camada PHY, são elas: *Radio Frequency (RF)*, *PHY Data (PD)* e *PHY Layer Management Entity (PLME)*.

O RF é constituído de parte de *hardware* e *firmware*, assim pode-se dizer que é físico e lógico, enquanto o PD e PLME são puramente interfaces lógicas, pois provem dados e gerenciamento de acesso aos serviços entre a camada PHY e o canal de radiofrequência. O modelo de referência da camada PHY é mostrado na Figura 7.

Figura 7: Modelo de referência da camada PHY



Fonte: Adaptado de IEEE STD802.15.4, 2003.

As Unidades de Dados de Protocolo MAC – *MAC Protocol Data Unit (MPDU)* são transportadas da camada MAC para a PD através do seu Ponto de Acesso ao Serviço – PD-SAP. Os pacotes de dados do IEEE 802.15.4 são de tamanho máximo de 127 bytes.

Os comandos de gerenciamento entre a camada MAC e o PLME são realizados através de primitivas suportadas pela entidade PLME-SAP.

O PLME tem também a responsabilidade de manter um banco de dados de objetos gerenciados pelo PHY chamado *PHY PAN Information Base* (PHY PIB).

A interface destas entidades com o canal de RF é realizada através do RF-SAP.

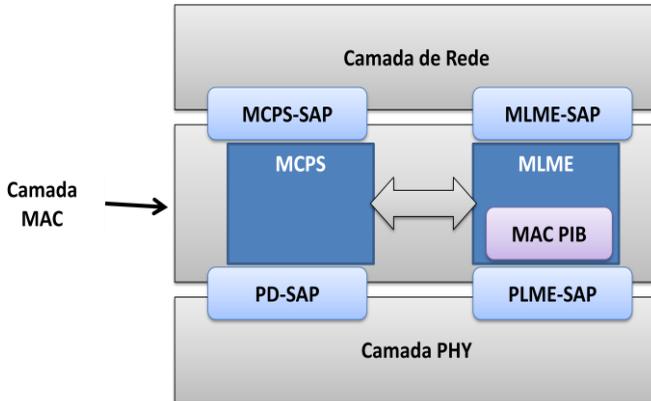
2.2.2 Camada MAC

Segundo Daintree, 2008, a camada MAC (*Medium Access Control*) é responsável por prover comunicação confiável entre um nó e seu vizinho imediato na rede, ajudando a evitar colisões e aumentar a eficiência nas transmissões. A camada MAC ainda é responsável pelos serviços de associação e dissociação da rede.

Ferdinando, 2005, comenta que a camada MAC controla o acesso ao canal de rádio compartilhado pelos nós da rede. Esta camada gera e reconhece os endereços da rede, e verifica as sequencias das estruturas de controle (*frame check*), ou seja, é responsável pelo processo do encapsulamento dos dados vindo das camadas superiores preparando-os para serem transmitidos pelo canal de radio e vice versa.

Os serviços da camada MAC são providos através de duas entidades: *MAC Commom Part Sub-Layer* (MCPS) e MAC Layer Management Entity (MLME), que são descritos nos manuais (802154MPSUG, 2006) e (802154MPSRM, 2006). A Figura 8 mostra o modelo de referência para a camada MAC.

Figura 8: Modelo de referência da camada MAC.



Fonte: Adaptado de IEEE STD802. 15.4, 2003.

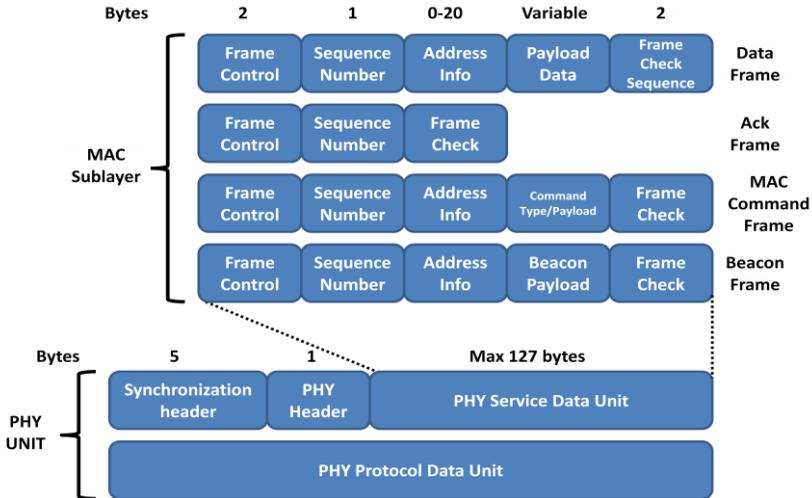
As unidades de Dados do Protocolo de Rede – SPDU são transportadas da camada de rede para a entidade MCPS através de seu Ponto de Acesso ao Serviço - MCPS-SAP.

Os comandos de gerenciamento entre a camada de Rede e o MLME da camada MAC são realizados através das primitivas suportadas pela entidade MLME-SAP.

O protocolo *Zigbee* permite que sejam implementados diferentes formas para o tráfego de dados. Podem existir aplicações que requeiram tráfego de dados periódicos, a exemplo de aplicações com sensores, outras aplicações podem demandar tráfego de dados de forma intermitente caso, por exemplo, de aplicações de automação residencial como controles remoto e também existem aplicações onde se faz necessário o tráfego de dados repetitivos com baixa latência caso, por exemplo, de uma aplicação como um *mouse*.

Para que isso seja possível, a camada MAC trabalha com quatro diferentes tipos de mensagens, que variam dependendo das necessidades da rede, com formatos conforme mostrado na Figura 9. Nesta figura pode-se verificar que dependendo do tipo da mensagem o número de *bytes* enviados varia, deixando a mensagem mais curta em alguns casos diminuindo assim o tráfego de dados da rede. Para que seja possível receber e tratar a mensagem adequadamente, o *PHY Header* é um *byte* dedicado a indicar o tamanho da mensagem.

Figura 9: Estrutura das mensagens na camada MAC



Fonte: Adaptado de GALEEV, 2004.

Os quatro tipos de mensagens da camada MAC são:

- *Beacon*, que é uma mensagem de sinalização/sincronismo, a qual é enviada pelos roteadores da rede em períodos pré-determinados e serve para que os demais nós, exceto o coordenador da rede, se programem para ao acabar sua tarefa entrem em estado de baixo consumo, voltando a atividade apenas na ocorrência de uma nova mensagem de sinalização. Para fazer a sincronização o método CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*) é utilizado. Este método faz com que um nó verifique a ausência de tráfego antes de iniciar uma transmissão, com o objetivo de diminuir a possibilidade de transmissões simultâneas. Este procedimento garante que o nó da rede só inicie a transmissão depois que o canal seja encontrado e determinado como livre. Em algumas aplicações, onde o consumo de energia não é um problema, pode ser utilizada comunicação direta, sem a necessidade de utilizar as mensagens de *beacon*;

- *Command frame*, é a estrutura utilizada quando é necessário enviar um comando do protocolo Zigbee a outro dispositivo da rede. Estas estruturas são bem definidas e estão descritas nas especificações disponibilizadas pela Zigbee Alliance;

- *Ack frame*, ou *acknowledgement frame* é uma mensagem enviada por nó que recebeu uma mensagem para o nó que enviou esta mensagem, confirmando o recebimento de uma mensagem válida;

- *Data frame* é a mensagem utilizada para realizar o envio de dados entre os dispositivos da rede. Os dados enviados podem ter tamanhos variáveis, se ajustando a necessidade da aplicação. Cada mensagem é destinada a um endereço da rede.

2.2.3 Camada de rede (NWK)

A camada de rede, ou NWK, faz a interface entre as camadas MAC e de aplicação e é a camada responsável por fazer a gestão rede, armazenamento de endereços, e roteamento de mensagens.

Segundo ADAMS, 2009 todos os dispositivos descritos pelo IEEE 802.15.4 possuem um endereço de rede de 64 bits único na rede, chamado de endereço longo, ou *long address*. Este endereço longo é similar aos endereços utilizados nos protocolos de comunicação sem fios 802.11 ou de *ethernet* 802.3. Contudo, em redes complexas que transmitem pequenos blocos de informação, pode-se utilizar um endereço de rede reduzido (*short address*) com 16 bits, que pode ser utilizado localmente. Esta troca de endereços permite que as comunicações entre a rede seja realizada de forma mais eficiente e com pacotes de transmissões menores. O coordenador da rede é o responsável por estabelecer um endereço de rede reduzido a um dispositivo que se une a esta.

A Tabela 3 mostra como é feita a identificação dentro do protocolo *Zigbee*. Podemos perceber que tudo começa com um canal físico de radiofrequência, conforme descrito na camada PHY. Neste canal o coordenador da rede inicia uma rede a qual recebe um número que a identifica chamado de PAN ID. Este endereço é exclusivo para cada canal da rede *Zigbee*.

Tabela 3: Range de endereçamento *Zigbee*

Nome	Range	Descrição
Canal	11-26	Canal de radiofrequência
PAN ID	0x0000-0x3fff	Endereço de uma rede em um canal
NwkAddr	0x0000-0xffff7	Endereço de um nó em uma

		rede
Endpoint	1-240	Endereço de uma aplicação em um nó
Cluster	0x0000-0xffff	Objeto em uma aplicação
Command	0x00-0xff	Uma ação a ser feita em um cluster
Attribute	0x0000-0xffff	Um dado em um <i>cluster</i>

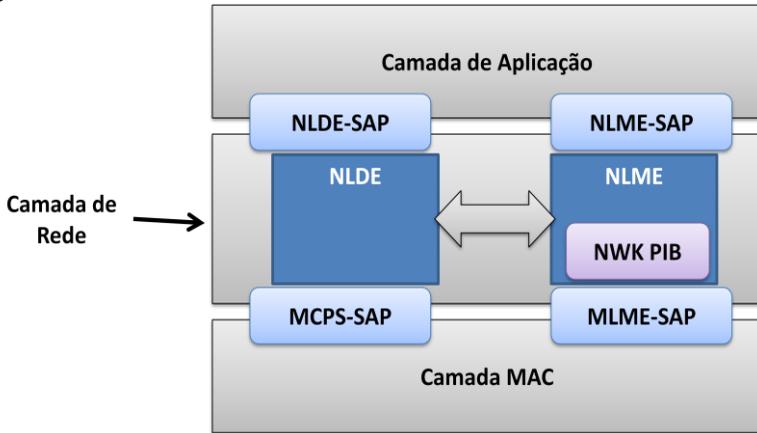
Fonte: Adaptado de GISLASON, 2008.

Os dispositivos ou nós da rede são identificados através de um endereço também único chamado *network address* (NwkAddr). Cada nó pode ter de 1 a 240 aplicações, ou *endpoints*. Para facilitar o agrupamento e a comunicação entre as aplicações o protocolo *Zigbee* utiliza estruturas chamadas *clusters*, que nada mais são que objetos de uma aplicação, como por exemplo, em uma aplicação de acendimento de uma lâmpada podemos ter dois *clusters*, um que representa um interruptor e outro que aciona a lâmpada.

Cada *cluster* é formado por um grupo de comandos e atributos o qual o deixa único e preparado para trocar informações com outros clusters pré-definidos.

A Figura 10 mostra o modelo de referência da camada de rede. Nela podemos perceber que os serviços são providos por duas entidades: *NWK Layer Data Entity* (NLDE) e *NWK Layer Management Entity* (NLME).

Figura 10: Modelo de referência da camada de rede



Fonte: Adaptado de IEEE STD802. 15.4, 2003.

As Unidades de Dados do Protocolo de Aplicação – *Application Protocol Data Unit* (APDU) são transportadas da camada de aplicação para a entidade NLDE através do seu Ponto de Acesso ao Serviço NLDE-SAP.

Os comandos de gerenciamento entre a camada de aplicação e o NLME da camada de rede são realizados através das primitivas suportadas pela entidade NLME-SAP.

2.2.4 Camada de segurança

FARAHANI (2008) comenta que em uma rede sem fios, as mensagens transmitidas podem ser recebidas por qualquer dispositivo que esteja a uma distância alcançável, inclusive um dispositivo intruso. Com isso existem duas preocupações principais com relação a segurança em redes sem fios.

A primeira delas é da confidencialidade da informação. O dispositivo intruso pode ter acesso a informações importantes apenas por estar recebendo as mensagens que estão sendo transmitidas. Criptografar as mensagens antes do envio resolve o problema de confidencialidade. Este processo modifica a mensagem utilizando uma combinação de bits conhecida como chave de segurança, a qual, só pode ser recuperada pelo dispositivo o qual a mensagem foi destinada. O padrão IEEE 802.15.4 suporta o uso do Padrão Avançado de Criptografia (AES)

A segunda preocupação é que o dispositivo intruso possa modificar e reenviar uma mensagem, mesmo que a mensagem esteja criptografada. Este problema normalmente é resolvido com a adição de um código de integridade o qual é conhecido como autenticação de dado e permite que o dispositivo que receber a mensagem verifique se ela foi modificada durante o seu envio.

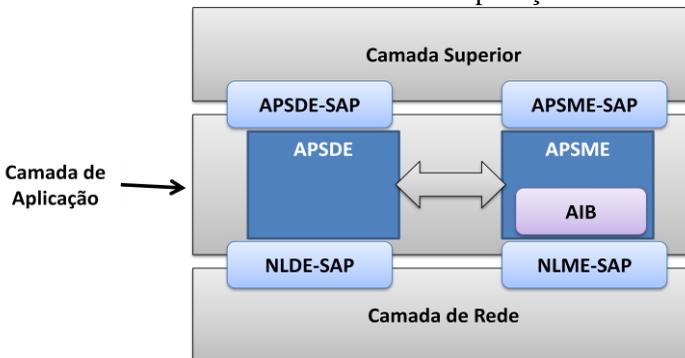
Uma das maiores dificuldades em se implementar elementos de segurança nas redes de comunicação *Zigbee* são os recursos limitados dos dispositivos da rede, os quais são normalmente projetados para operar a bateria, com processamento e memória limitados.

2.2.5 Camada de aplicação

A camada de aplicação é composta de acordo com *Zigbee Specification* (2006) pela Subcamada de Suporte de Aplicação (*Application Support Sub-Layer - APS*), pelos Objetos dos Dispositivos *Zigbee - Zigbee Devices Objects (ZDO)* e pelos Objetos de Aplicação definidos pelos fabricantes e hospedados *Framework* de Aplicações – *Application Framework (AF)*.

A Figura 11 mostra o modelo de referência descrito para a camada de aplicação. Nela podemos perceber que as camadas superiores da rede, definidas pelo usuário final, se comunicam com a APSDE, para o envio de pacotes de dados através da APSDE-SAP, e com a APSME para o envio de comandos e gerenciamento através da APSME-SAP.

Figura 11: Modelo de referência da camada de aplicação



Fonte: Adaptado de IEEE STD802. 15.4, 2003.

As camadas de suporte a aplicação e de aplicação são as

camadas no topo do protocolo Zigbee. Os fabricantes desenvolvem objetos de aplicação para customizar um dispositivo para varias aplicações diferentes. Os objetos de aplicação controlam e manejam as demais camadas do protocolo em um dispositivo Zigbee. Pode-se ter até 240 objetos de aplicação em um único dispositivo Zigbee.

O objeto de aplicação zero, também chamado de ZDO, é responsável por implementar o perfil do dispositivo e também é o gerente da rede.

Para facilitar a implementação, o padrão *Zigbee* oferece a opção de uso de perfis de aplicação. Estes perfis são uma série de acordos e padronizações feitas determinando funcionalidades específicas para um certo tipo de aplicação. O uso de objetos de aplicação permite que os dispositivos sejam interoperáveis, ou seja, que dispositivos de diferentes fabricantes interajam entre si em uma rede. Como exemplo de perfis de aplicação podem ser citados o “*Home Automation*” mostrado na Tabela 4 e o “*Smart Energy*”.

Cada um destes perfis disponibiliza dispositivos e estruturas padrões as quais podem ser utilizadas de forma a minimizar o tempo de desenvolvimento de novas aplicações. Pode-se perceber nesta tabela que existem endereços descritos como reservados, os quais podem ser customizados pelo projetista do sistema para novas aplicações que não se enquadram nas opções disponíveis.

Tabela 4: Perfil de aplicação *Zigbee* “*Home Automation*”

	Device	Device ID		Device	Device ID	
Generic	On/Off Switch	0x0000	HVAC	Heating/Cooling Unit	0x0300	
	Level Control Switch	0x0001		Thermostat	0x0301	
	On/Off Output	0x0002		Temperature Sensor	0x0302	
	Level Controllable Output	0x0003		Pump	0x0303	
	Scene Selector	0x0004		Pump Controller	0x0304	
	Configuration Tool	0x0005		Pressure Sensor	0x0305	
	Remote Control	0x0006		Flow Sensor	0x0306	
	Combined Interface	0x0007		Reserved	0x0307 - 0x3FF	
	Range Extender	0x0008		Intruder Alarm Systems	IAS Control and Indicating Equipment	0x0400
	Mains Power Outlet	0x0009			IAS Ancillary Control Equipment	0x0401
	Door Lock	0x000A	IAS Zone		0x0402	
	Door Lock Controller	0x000B	IAS Warning Device		0x0403	
	Simple Sensor	0x000C	Reserved	0x0404 - 0xFFFF		
	Reserved	0x000C - 0x00FF	Closures	Shade	0x0200	
	Lighting	On/Off Light		0x0100	Shade Controller	0x0201
		Dimmable Light		0x0101	Window Covering Device	0x0202
		Color Dimmable Light		0x0102	Window Covering Controller	0x0203
On/Off Light Switch		0x0103		Reserved	0x0204 - 0x2FF	
Dimmer Switch		0x0104				
Color Dimmer Switch		0x0105				
Light Sensor		0x0106				
Occupancy Sensor		0x0107				
Reserved		0x0108 - 0x1FF				

Fonte: produção do próprio autor.

As estruturas, também chamadas de *clusters*, são especificações que cada tipo de dispositivo dentro de um perfil de aplicação devem ter. Novamente no caso do perfil de aplicação “*Home Automation*”, temos disponíveis para uso os *clusters* mostrados na Tabela 5. Dependendo do tipo de dispositivo alguns destes clusters são obrigatórios e outros opcionais. Além disso, alguns tem função de entrada e outros de saída, onde são complementares em determinadas aplicações.

Tabela 5: Clusters disponíveis na aplicação “*Home Automation*”

Functional Domain	Cluster Name	Cluster ID	Functional Domain	Cluster Name	Cluster ID
General	Basic	0x0000	Lighting	Color Control	0x0300
General	Power Configuration	0x0001	HVAC	Pump Configuration and Control	0x0200
General	Device Temperature Configuration	0x0002	HVAC	Thermostat	0x0201
General	Identify	0x0003	HVAC	Fan Control	0x0202
General	Groups	0x0004	HVAC	Thermostat User Interface Configuration	0x0204
General	Scenes	0x0005	Closures	Shade Configuration	0x0100
General	On/Off	0x0006	Closures	Door Lock	0x0101
General	On/Off Switch Configuration	0x0007	Closures	Window Covering	0x0102
General	Level control	0x0008	Security and Safety	IAS ACE	0x0501
General	Alarms	0x0009	Security and Safety	IAS Zone	0x0500
General	Binary Input (Basic)	0x000F	Security and Safety	IAS WD	0x0502
Measurement & Sensing	Illuminance Measurement	0x0400	Smart Energy	Meter	0x0702
Measurement & Sensing	Illuminance Level Sensing	0x0401			
Measurement & Sensing	Temperature Measurement	0x0402			
Measurement & Sensing	Pressure Measurement	0x0403			
Measurement & Sensing	Flow Measurement	0x0404			
Measurement & Sensing	Relative Humidity Measurement	0x0405			
Measurement & Sensing	Occupancy sensing	0x0406			

Fonte: produção do próprio autor.

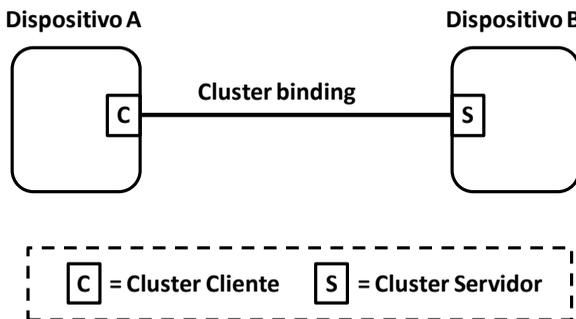
Outros perfis de aplicação disponibilizam clusters diferentes dos apresentados acima, deixando estruturas dedicadas e melhor adaptadas a realidade de cada aplicação.

Um conceito importante no protocolo *Zigbee* é o *binding*. Esta função permite a criação de *links* lógicos entre aplicações que estão relacionadas entre diferentes dispositivos da rede, como no caso de clusters do mesmo tipo, mas com função de entrada ou de saída, ou seja, complementares. Por exemplo, pode-se citar um dispositivo *Zigbee* conectado a uma lâmpada, o qual deve ser conectado logicamente a um dispositivo com capacidade de atuar como uma chave seletora, que irá controlar o estado da lâmpada. A informação relacionada a estes links lógicos é armazenada na chamada *binding table* que é gerida pela camada de aplicação.

Para a conexão ou *binding* dos *clusters* é utilizado um modelo de cliente/servidor. Este modelo é ilustrado na Figura 12. Tipicamente a entidade que armazena os atributos de um *cluster* é considerada como servidor e a entidade que afeta ou manipula este atributo é considerada

como cliente. Comandos que permitem que dispositivos manipulem atributos, como por exemplo, os comandos *Read Attribute* e *Write Attribute* são tipicamente enviados pelo dispositivo cliente e recebidos pelo dispositivo servidor. Qualquer resposta a estes comandos, como no caso dos comandos *Read Attribute Response* e *Write Attribute Response* são enviados do dispositivo servidor para o dispositivo cliente.

Figura 12: Modelo cliente servidor para a conexão de *clusters*



Fonte: Adaptado de *Zigbee Cluster Library Specification*, 2008.

Os *cluster* suportados por um perfil de aplicação são identificados através da estrutura “Simple Descriptor”, especificada em cada *endpoint* de um dispositivo. No “Simple Descriptor”, a lista de *clusters* de entrada deve conter a lista de *clusters* servidores suportadas pelo dispositivo e a lista de *clusters* de saída deve conter a lista de *clusters* clientes suportados pelo dispositivo.

2.3 KIT DE DESENVOLVIMENTO

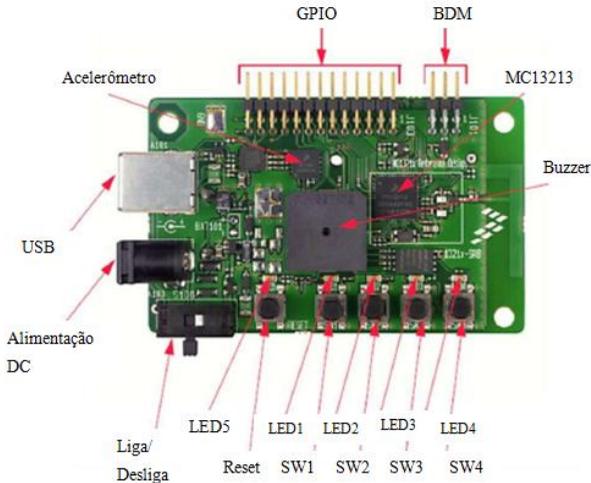
O *kit* de desenvolvimento utilizado para a implementação da rede *Zigbee* deste trabalho é o 13213-*Sensor Reference Board* (SRB) da Freescale (MC1321xEVKRM, 2008). Este *kit* é composto pelo *transceiver* de radio frequência MC13213, que é compatível com o protocolo IEEE802.15.4 e opera na faixa de frequência de 2,4 GHz.

Este *transceiver* inclui ainda um amplificador de baixo ruído, uma potência nominal de saída de 1 mW, um oscilador controlado por tensão (VCO), chave integrada para transmissão e recepção e um regulador de tensão interno. Na mesma pastilha do *transceiver* está um microcontrolador da família HCS08, o qual possui 64 KB de memória flash e 4 KB de memória RAM. Tem 8 canais multiplexados em um

conversor A/D de 10 bits e 56 I/Os digitais. É encapsulado em uma pastilha LGA com 71 pinos. A alimentação do circuito é feita com uma tensão de 3,3 VDC.

A Figura 13 mostra o *kit* de desenvolvimento utilizado indicando cada circuito na placa de circuito impresso.

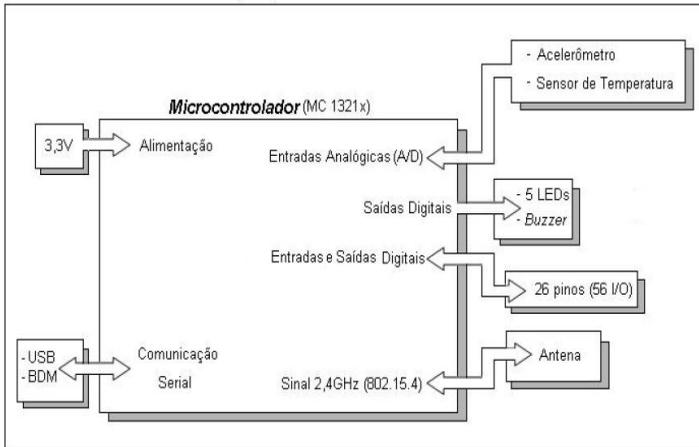
Figura 13: *Kit* de desenvolvimento 13213-SRB



Fonte: Adaptado de MC1321xEVKRM, 2008.

Já a Figura 14 mostra o diagrama de blocos do *kit* de desenvolvimento, deixando claro o fluxo de informações possível com o processador.

Figura 14: Diagrama de blocos do *kit* de desenvolvimento MC13213 SRB



Fonte: Adaptado de 13213EVKUG, 2007.

Além do *transceiver* o *kit* conta ainda com:

- Entradas e saídas digitais: O microcontrolador MC13213 possui 56 I/Os digitais que podem ser configuradas como entradas ou saídas. Algumas destas I/Os já são utilizadas por elementos do módulo para fazer a interface com o usuário, que é composta de 5 LEDs e 5 chaves, com utilização configurável conforme a aplicação. O módulo também oferece um conector de 26 pinos, chamado de GPIO para se configurar as I/Os de acordo com a necessidade de cada aplicação. Uma das saídas digitais é também destinada para chavear um circuito de acionamento de um *buzzer*, para emitir sinais sonoros, quando necessário.
- Entradas analógicas: O módulo possui um conversor A/D que trabalha com 8 entradas multiplexadas. Duas destas entradas são dedicadas a um sensor de temperatura e um acelerômetro, para aplicações que necessitem medir variações de velocidade ou deslocamento, presentes no próprio módulo. O módulo não dispõe de conversor D/A, sendo assim não é possível se implementar saídas analógicas.

- Porta USB 2.0: Usado como dispositivo de comunicação para a transferência de dados e também para ajudar na depuração de aplicações através de um PC.
- Antena tipo F para a transmissão sem fio em 2,4GHz vem confeccionada na própria placa de circuito impresso do módulo.
- Modulo de *debug* (*Background Debug Module* - BDM): Serve como alternativa para a gravação de programas no microprocessador e também para depuração. É conectada ao PC pela porta USB.
- Porta baterias (2xAA): Permite a utilização do *kit* em aplicações móveis.

O ambiente de desenvolvimento utilizado para a programação do microcontrolador foi o *Code Warrior* 6.3. Este *kit* de desenvolvimento foi escolhido para o desenvolvimento deste trabalho por atender as necessidades da aplicação e por estar disponível no laboratório de sistemas embarcados em boa quantidade.

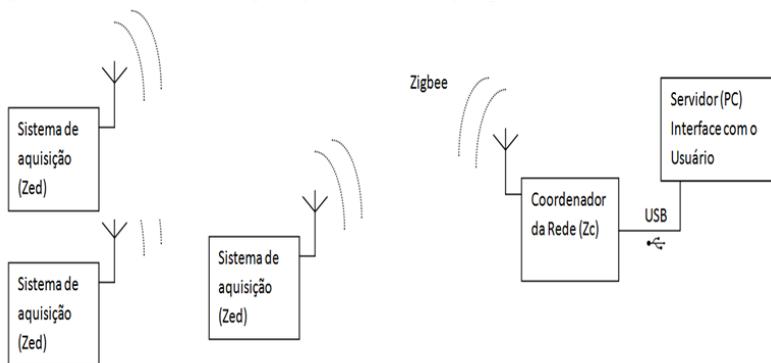
2.4 IMPLEMENTAÇÃO DA REDE SEM FIO PARA AQUISIÇÃO DE SINAIS

O objetivo deste trabalho é a implementação de uma rede sem fios para a aquisição de sinais bioelétricos utilizando o protocolo *Zigbee*. Inicialmente o sistema irá adquirir sinais de oxigenação do sangue, batimentos cardíacos e temperatura corporal. O sistema proposto é mostrado na Figura 15 e é constituído por dispositivo coordenador da rede *Zigbee*, o qual se comunica via USB com um computador, que roda o *software* de interface com o usuário, o qual foi desenvolvido em Labview permite a visualização dos dados que estão sendo adquiridos pelos sensores da rede *Zigbee*.

O dispositivo coordenador da rede é responsável por formar a rede e por detectar os outros dispositivos sensores presentes nela. Periodicamente o coordenador envia mensagens pedindo os dados adquiridos pelos dispositivos sensores e assim que recebe a mensagem com os dados faz o envio ao computador para que sejam disponibilizados para o usuário do sistema.

Com os dados no computador o usuário é capaz de visualizar gráficos para facilitar a interpretação dos dados coletados além de poder armazená-los para visualização posterior.

Figura 15: Sistema de aquisição de dados proposto



Fonte: Produção do próprio autor.

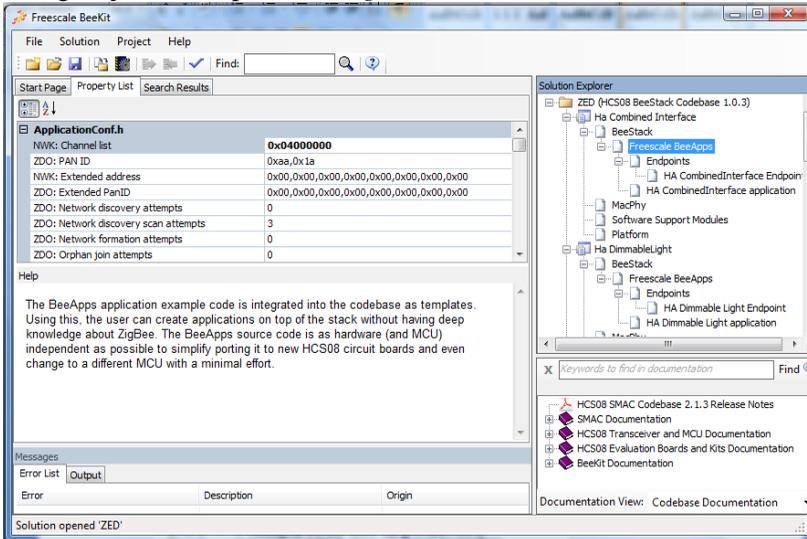
Para simplificar esta implementação e devido ao fato de termos um número limitado de dispositivos, foi utilizada uma topologia de rede em estrela, onde se tem um dispositivo coordenador da rede e vários dispositivos *end devices*. Steinhauer (2012), comenta que esse tipo de rede não tem muito alcance, pois não possui roteador, e por isso também, não se deve utilizar em ambientes com muitos obstáculos. Outras topologias poderiam ser facilmente adotadas, como a topologia árvore ou malha, precisando apenas configurar os dispositivos para tal.

Para configuração dos dispositivos foi utilizado o *software* BEEKIT, disponibilizado pela Freescale para o desenvolvimento de aplicações utilizando o protocolo *Zigbee*. O documento BKWCTKUG, 2011, foi utilizado como referência para a utilização deste *software*. A Figura 16 mostra a interface principal do *software* BEEKIT. Este *software* disponibiliza uma série de opções pré-configuradas o que permite agilidade na implementação de novas redes utilizando o protocolo *Zigbee* e os *kits* de desenvolvimento da Freescale.

Alguns exemplos de parâmetros configuráveis através do BEEKIT, são o canal de rádio a ser utilizado, o *kit* de desenvolvimento a ser utilizado e a habilitação ou não de módulos disponíveis. Configurações de endereço de rede e de dispositivos, além de opções de autenticação de dispositivos.

Outro ponto importante é a configuração da camada de aplicação. Já estão pré-disponibilizados vários *clusters* que podem ser utilizados para operações básicas dos dispositivos, como aquisição de sinais e outros atuadores, como por exemplo no caso de acendimento de lâmpadas ou em funções de termostatos.

Figura 16: Janela de apresentação do *Software* Beekit da Freescale para configuração dos dispositivos *Zigbee*



Fonte: Produção do próprio autor.

2.4.1 Configuração dos dispositivos

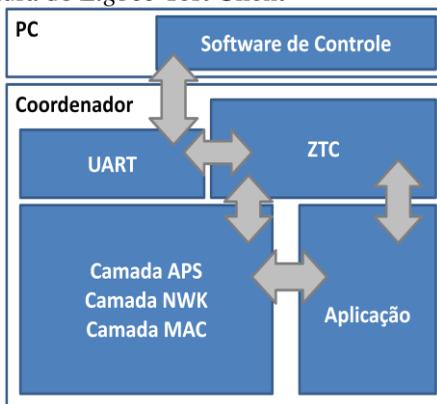
O coordenador da rede foi configurado de forma a serem habilitados os módulos de teclado e de led's, os quais são essenciais para que o usuário possa interagir com os dispositivos. Além disso, foi habilitado o módulo de comunicação UART via USB disponível nas placas do *kit* de desenvolvimento e junto com este módulo, também foi configurado o uso do *Zigbee Test Client* (ZTC) o qual é um módulo do protocolo *Zigbee* que permite testar os dispositivos presentes na rede além de inspecionar as mensagens trocadas via rede sem fio através de um *software* externo de monitoramento. O ZTC é habilitado atribuindo o valor *True* a propriedade *ZTC:ZTC enabled* no *Software Support Modules* do BEEKIT.

O ZTC funciona conforme a estrutura mostrada na Figura 17 e

só é aplicável para o coordenador da rede. Nesta figura pode-se perceber que o módulo ZTC tem interface com todas as camadas do protocolo *Zigbee* (MAC, NWK, APS) e com a aplicação do usuário. Este módulo compila as informações e se comunica via UART com um *software* externo, o qual disponibiliza para o usuário as informações da rede.

A interface com o usuário desenvolvida neste trabalho é melhor explicada no capítulo 4. Ela utiliza o ZTC com forma de comunicação com a rede *Zigbee*.

Figura 17: Estrutura do *Zigbee* Test Client



Fonte: Adaptado de BSBZTCRM, 2011.

O coordenador foi configurado para gerar um endereço de rede randômico e para utilizar o canal de comunicação número 14. A escolha do canal de comunicação utilizado nesta aplicação foi arbitrária e conforme mostrado na Figura 6, poderiam ser utilizados qualquer canal entre os canais 11 e 26, pois está se utilizando a frequência de 2,4 GHz para a transmissão dos dados.

Para o coordenador as SAPs a seguir devem ser configuradas para que seja possível receber e transmitir comandos para a interface com o usuário no computador (as demais SAPs podem ser desabilitadas):

- ZDP SAP (habilitada atribuindo *True* para a propriedade *ZTC: ZDP SAP Capability* no *Software Support Modules* do BEEKIT)
- APSDE SAP (habilitada atribuindo *True* para a propriedade *ZTC: APSDE SAP Capability* no *Software Support Modules* do BEEKIT)
- APSME SAP (habilitada atribuindo *True* para a propriedade

ZTC: APSME SAP Capability no *Software Support Modules* do BEEKIT)

- NLME SAP (habilitada atribuindo *True* para a propriedade *ZTC: NLME SAP Capability* no *Software Support Modules* do BEEKIT)

Grandes pacotes de dados são trocados entre a interface com o usuário e o dispositivo coordenador da rede, com isso para garantir uma comunicação confiável, deve-se habilitar o controle de fluxo por hardware, tanto no *software* embarcado no dispositivo coordenador, quanto no *software* de interface com o usuário em Labview.

Para o dispositivo coordenador esta propriedade é habilitada atribuindo *True* para *UART 2 Hardware flow control enabled* na aba *Platform* do BEEKIT. Como consequência do uso de controle de fluxo as propriedades *UART RX Flow Control Skew* e *UART RX Flow Control Resume* da aba *Platform* do BEEKIT devem ser colocadas em 10.

Para que seja possível a verificação correta das características de cada dispositivo da rede via interface com o usuário, o coordenador precisa das propriedades abaixo habilitadas.

- *Active Endpoint request* (habilitada atribuindo *True* para a propriedade *ZDP: Enable Active_EP_req* no Freescale BeeApps do BEEKIT)

- *Simple Descriptor request* (habilitada atribuindo *True* para a propriedade *ZDP: Enable Simple_Desc_req* no Freescale BeeApps do BEEKIT)

- *Node Descriptor request* (habilitada atribuindo *True* para a propriedade *ZDP: Enable Node_Desc_req* no Freescale BeeApps do BEEKIT)

- *Power Descriptor request* (habilitada atribuindo *True* para a propriedade *ZDP: Enable Power_Desc_req* no Freescale BeeApps do BEEKIT)

- *Management Bind request* (habilitada atribuindo *True* para a propriedade *ZDP MGMT: Enable Mgmt_Bind_req* no Freescale BeeApps do BEEKIT)

- *Management Routing request* (habilitada atribuindo *True* para a propriedade *ZDP MGMT: Enable Mgmt_Rtg_req* no Freescale BeeApps do BEEKIT).

Todos os dispositivos conectados a rede ainda precisam ter as propriedades abaixo habilitadas.

- *MGMT_Bind response* – Necessário para responder ao

comando `MGMT_Bind request`. Com isso a interface com o usuário é informada sobre as conexões entre os dispositivos na rede. Para habilitar esta propriedade é necessário atribuir *True* para a propriedade `ZDP MGMT: Enable Mgmt_Bind_rsp` na aba Freescale BeeApps do BEEKIT.

- `MGMT_Rtg response` – Necessário para responder ao comando `MGMT_Rtg request`. Para habilitar esta propriedade é necessário atribuir *True* para a propriedade `ZDP MGMT: Enable Mgmt_Rtg_rsp` na aba Freescale BeeApps do BEEKIT.

- Deve-se atribuir *True* também a propriedade `ZDP: enable End_Device_Annce` no Freescale BeeApps do BEEKIT.

- A propriedade `NWK: Maximum Broadcast Transaction Table Entries` do Freescale BeeApps deve ser setado em 10 para todos os dispositivos.

Nos dispositivos sensores, configurados como *end devices* para a topologia da rede *Zigbee*, foram também habilitados os módulos de teclado e de led's, mas não se fez necessário o uso do módulo UART. Para os fins acadêmicos deste trabalho não foi implementada a função de baixo consumo de energia, contudo caso seja necessário implementação o *software* foi escrito de forma a suportá-la.

Para que seja possível a comunicação com o coordenador da rede, os dispositivos sensores também foram configurados para trabalhar utilizando o canal 14 de comunicação.

Os dispositivos sensores também precisam de configurações especiais, visto que devem fornecer informações ao coordenador da rede e a interface com o usuário.

- A propriedade `MAC Rx On When Idle` deve ser habilitada atribuindo o valor *True* ao `MAC: Mac Capability: Rx On When Idle` na aba Freescale BeeApps do BEEKIT.

- A propriedade `ZDO: Maximum number of Poll failure before rejoin` da aba Freescale BeeApps deve ser configurada como 5.

A camada de aplicação foi configurada utilizando o perfil de aplicação *“Home Automation”*. Este perfil foi escolhido por ser compatível com o *kit* de desenvolvimento utilizado neste trabalho e disponível na UDESC, além de atender as necessidades deste projeto com os *clusters* que disponibiliza.

O perfil de aplicação *“ZigBee Health Care”*, disponível dentro das especificações da *Zigbee Alliance*, e mais indicado para o uso neste trabalho visto que é dedicado para sistemas biomédicos, não pode ser

utilizado pois requer o uso de um microcontrolador mais elaborado (ARM7, ao invés de HCS08 presente nos *kits* MC13213), o que acarretou na necessidade de aquisição de novos *kits* de desenvolvimento. Com isso optou-se pela adaptação do perfil “*Home Automation*” para a aquisição de sinais bioelétricos, aproveitando assim os *kits* de desenvolvimento disponíveis na UDESC. O trabalho realizado nesta implementação pode ser facilmente convertido para o uso do protocolo do “*ZigBee Health Care*”, sendo necessário apenas reconfigurar a camada de aplicação.

Dentro do perfil de aplicação “*Home Automation*”, o dispositivo coordenador da rede foi configurado como sendo uma adaptação do tipo “*Combined Interface*” (device ID 0x0007). Este tipo de dispositivo tem como objetivo controlar e monitorar outros dispositivos e a rede em si e normalmente é alimentado através da rede elétrica e é conectado a um computador. Desta forma após pequenas adaptações se enquadrou perfeitamente nesta aplicação .

Os *clusters* inseridos no dispositivo “*Combined Interface*” são mostrados na Tabela 6.

Tabela 6: *Clusters* do dispositivo coordenador da rede

Combined Interface – Coordenador		
Nome do Cluster	Cluster ID	Tipo
Basic	0x0000	Entrada
Identify	0x0003	Entrada
Groups	0x0004	Entrada
Temperature Measurement	0x0402	Entrada
Oxímetro de pulso	0x0407	Entrada
Batimentos Cardíacos	0x0408	Entrada

Fonte: Produção do próprio autor.

Os dispositivos sensores da rede foram configurados como adaptações do tipo “*Temperature Sensor*” (device ID 0x0302). Este dispositivo é capaz de fazer a aquisição de dados e enviar este sinal para outro dispositivo da rede. Esta característica foi então adaptada para a aquisição dos sinais biomédicos de interesse.

Os *clusters* presentes nos dispositivos sensores são os mostrados na Tabela 7, Tabela 8 e Tabela 9. Nota-se que todos os dispositivos apresentam os *clusters* “*Basic*”, “*Identify*” e “*Groups*”, estes *clusters* são padrões e servem para a identificação e gerenciamento

dos dispositivos na rede.

Tabela 7: *Clusters* do dispositivo sensor de temperatura corporal

Dispositivo - Sensor de Temperatura		
Nome do Cluster	Cluster ID	Tipo
Basic	0x0000	Entrada
Identify	0x0003	Entrada
Groups	0x0004	Entrada
Temperature Measurement	0x0402	Saída

Fonte: Produção do próprio autor.

Tabela 8: *Clusters* do dispositivo sensor de oxigenação no sangue

Dispositivo - Sensor Oxímetro de Pulso		
Nome do Cluster	Cluster ID	Tipo
Basic	0x0000	Entrada
Identify	0x0003	Entrada
Groups	0x0004	Entrada
Oxímetro de pulso	0x0407	Saída

Fonte: Produção do próprio autor.

Tabela 9: *Clusters* do dispositivo sensor de batimentos cardíacos

Dispositivo - Sensor Batimentos Cardíacos		
Nome do Cluster	Cluster ID	Tipo
Basic	0x0000	Entrada
Identify	0x0003	Entrada
Groups	0x0004	Entrada
Batimentos Cardíacos	0x0408	Saída

Fonte: Produção do próprio autor.

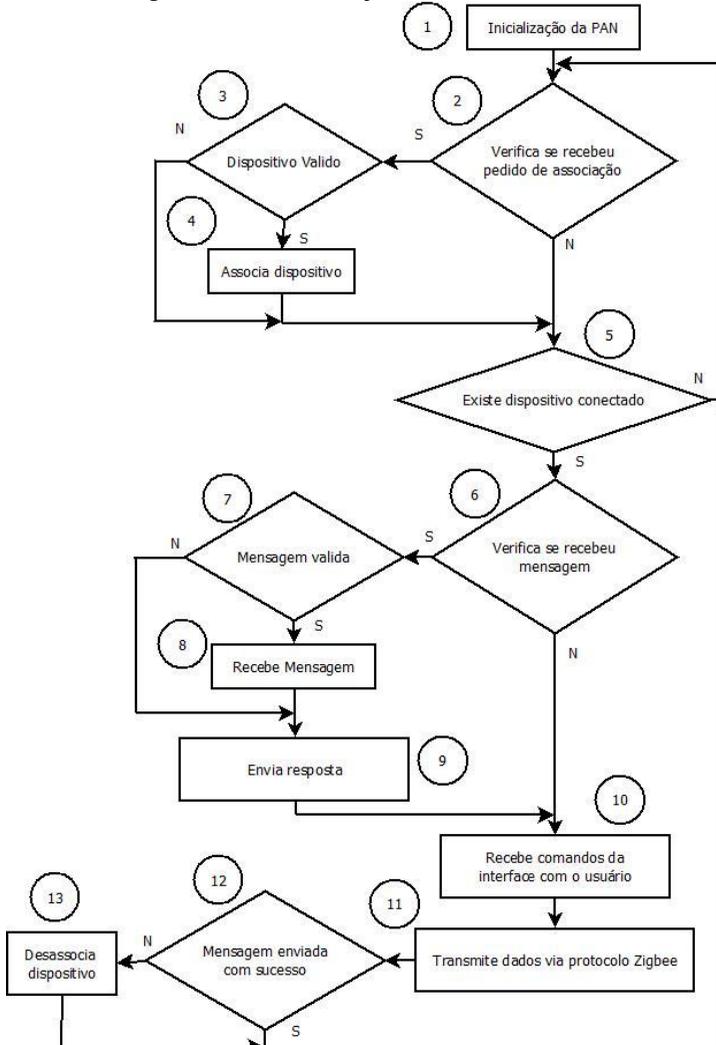
Inserindo estas informações no *software* BEEKIT, é gerada uma configuração básica para a utilização do protocolo *Zigbee*, que deve ser exportada para o *software* de desenvolvimento *Code Warrior*, no qual é possível realizar as customizações específicas necessárias para cada aplicação e onde o *software* do sistema embarcado é compilado.

Através do *Code Warrior*, foram configurados os módulos de aquisição de dados utilizados para a aquisição dos sinais bioelétricos descritos no capítulo 3.

No diagrama de blocos da Figura 18 pode se observar um

fluxograma resumido do *software* do coordenador, e a seguir, a Tabela 10 descreve as suas principais etapas. Deve-se lembrar de que muitas das funções pertinentes ao protocolo de transmissão sem fio estão representadas em alto nível e de forma apenas orientativa.

Figura 18: Fluxograma da inicialização do coordenador da rede.



Fonte: Produção do próprio autor.

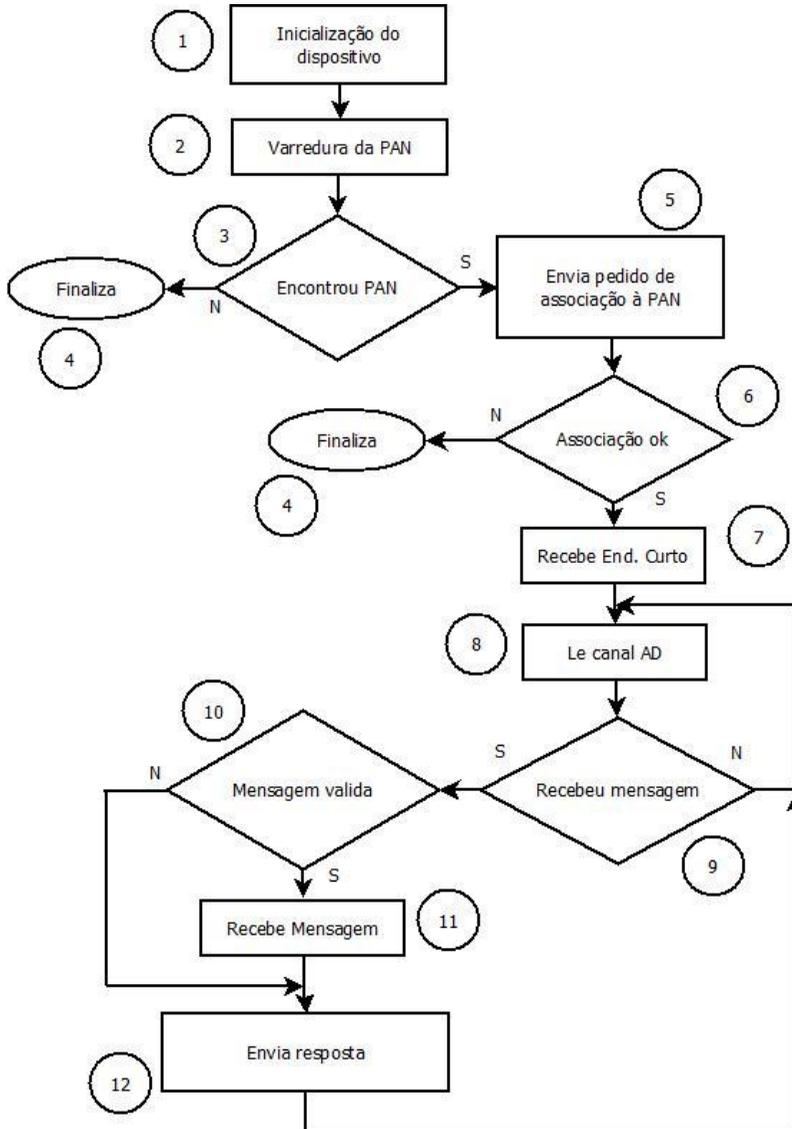
Tabela 10: Descrição das etapas de inicialização do coordenador da rede.

Etapa	Descrição
1	Nesta etapa o coordenador da rede executa toda a máquina de estados para a inicialização da PAN e define um canal, um PAN ID e um endereço curto para si.
2	O coordenador da rede verifica se existem pedidos de associação de novos dispositivos à rede.
3	O coordenador verifica se o dispositivo que enviou o pedido de associação é válido.
4	Caso o dispositivo seja valido ele é associado à rede e recebe o seu endereço de rede.
5	Caso existam dispositivos conectados a rede o coordenador verifica se recebeu mensagens, senão volta a verificar se existe pedidos de associação.
6	Cada mensagem enviada pela rede possui um identificador de destino e de origem, com isso é feita a verificação se a mensagem deve ou não ser recebida.
7	Analisa se a mensagem é valida ou não. Cada mensagem possui um <i>byte</i> de conferência o qual é a soma dos <i>bytes</i> anteriores. Isto é feito para garantir a recepção de mensagens íntegras.
8	Recebe a mensagem caso ela seja integra e prepara a resposta.
9	Envia a resposta para a mensagem recebida. Toda mensagem possui uma resposta.
10	Recebe os comandos do <i>software</i> de interface com o usuário, o qual gere o sistema de aquisição e verifica se alguma ação deve ser tomada.
11	Transmite os dados para os outros dispositivos da rede.
12	Verifica se a mensagem foi enviada com sucesso.
13	Caso não receba sinal de resposta para a mensagem enviada desassocia o dispositivo ao qual a mensagem era destinada.

Fonte: Produção do próprio autor.

No diagrama de blocos da Figura 19 pode se observar um fluxograma resumido do *software* do coordenador, e a seguir, a Tabela 11 descreve as suas principais etapas.

Figura 19: Fluxograma do *software* dos dispositivos finais (sensores) da rede.



Fonte: Produção do próprio autor.

Tabela 11: Descrição das etapas de inicialização dos dispositivos finais

(sensores) da rede.

Etapa	Descrição
1	Nesta etapa o dispositivo é inicializado.
2	O dispositivo faz uma varredura buscando encontrar um PAN em sua região de alcance.
3	Verifica se encontrou ou não uma PAN.
4	Se não encontrar uma PAN em um tempo determinado ou não conseguir se associar com outro dispositivo <i>software</i> é finalizado.
5	Envia um pedido de associação ao dispositivo coordenador da rede PAN encontrada.
6	Verifica se o pedido de associação foi aceito ou não pelo coordenador da rede.
7	Se for aceito na rede, a partir de seu endereço longo, recebe do coordenador da rede um endereço curto.
8	Realiza a leitura do canal AD ao qual está vinculado o seu sistema de medição biomédico.
9	Cada mensagem enviada pela rede possui um identificador de destino e de origem, com isso é feita a verificação se a mensagem deve ou não ser recebida.
10	Analisa se a mensagem é válida ou não. Cada mensagem possui um <i>byte</i> de conferencia o qual é a soma dos <i>bytes</i> anteriores. Isto é feito para garantir a recepção de mensagens integras.
11	Recebe a mensagem caso ela seja integra e prepara a resposta.
12	Envia a resposta para a mensagem recebida. Toda mensagem possui uma resposta.

Fonte: Produção do próprio autor.

2.4.2 INICIANDO A REDE

Para que se tenha maior flexibilidade para o controle da rede e para uma melhor utilização dos botões disponíveis no *kit* de desenvolvimento, foi desenvolvida uma rotina onde cada botão pode ser pressionado de duas formas diferentes, chamadas de “SW Curto” ou “SW Longo”. Apertar um botão por mais de um segundo caracteriza um “SW Longo”. Desta forma um botão pode ter duas funções diferentes programadas para ele. Os dispositivos iniciam em um modo de configuração e após iniciada a rede deve-se alterar para o modo de

execução.

Para iniciar a rede o usuário do sistema precisa:

- 1) Alimentar os dispositivos que farão parte da rede. Isto feito, o LED1 irá piscar indicando que aquela placa não está conectada a rede, caso ela seja um dispositivo *end device*, ou roteador, irá indicar que uma rede ainda não foi formada, no caso de um dispositivo coordenador.
- 2) Conectar o dispositivo coordenador da rede no computador via cabo USB.
- 3) Abrir o *software* “interface com o usuário” no computador e selecionar a porta de comunicação.
- 4) Pressionar SW1 no dispositivo coordenador da rede. Os leds LED1, LED2, LED3 e LED4 irão piscar consecutivamente até que apenas o LED1 permaneça aceso. Neste ponto a rede estará formada e a partir disso, o coordenador atribui um número identificador a todos os dispositivos que conectarem a rede.
- 5) Para adicionar outros nós sensores a rede pressione SW1 no dispositivo desejado. Os leds LED1, LED2, LED3 e LED4 irão piscar consecutivamente até que apenas o LED1 permaneça aceso. Neste ponto o dispositivo se juntou a rede.
- 6) O próximo passo é conectar as funções dos dispositivos, o que é chamado no protocolo *Zigbee* de *Binding*, conforme mostrado na seção 2.2.5 deste capítulo. Para isso, deve-se pressionar a SW3 do coordenador da rede e dentro de um intervalo de até dez segundos pressionar a SW3 do dispositivo nó sensor da rede o qual se deseja conectar. O LED3 deve piscar e ao parar e ficar aceso o processo de *binding* está finalizado com sucesso. Caso o LED3 apague após piscar o processo falhou e deve ser reiniciado.
- 7) Por último deve-se colocar o dispositivo em modo de aplicação realizando um pressionamento longo em SW1 em cada dispositivo.

O detalhamento das funções disponíveis em cada botão para o modo de configuração dos dispositivos está descrito abaixo:

- Pressionamento curto (menos de 1 segundo) de SW1 – Iniciar ou se juntar a uma rede;

- Pressionamento curto de SW2 (apenas no dispositivo coordenador) – Permite ou não que outros dispositivos se juntem a rede;
 - Junção permitida: LED2 ligado
 - Junção não permitida: LED2 desligado
- Pressionamento curto de SW2 – Sem função;
- Pressionamento curto de SW3 – Envia pedido de conexão (*binding*) para outro dispositivo da rede. O LED3 fica piscando por alguns segundos até que quando dois dispositivos se conectam o LED3 dos dois dispositivos param de piscar;
- Pressionamento longo (mais de 1 segundo) de SW1 – Alterna o dispositivo para o modo de aplicação. O dispositivo inicia no modo de configuração e ao se pressionar este botão o dispositivo muda de estado e executa a aplicação programada;
- Pressionamento longo de SW2 – O dispositivo deixa a rede;
- Pressionamento longo de SW3 – Remove todas as conexões (*binding*) entre os dispositivos da rede.

O detalhamento das funções disponíveis em cada botão para o modo de aplicação dos dispositivos está descrito abaixo:

- Pressionamento curto de SW1 – Faz a ação principal da aplicação;
- Pressionamento curto de SW2 – Sem função;
- Pressionamento curto de SW3 – Sem função;
- Pressionamento curto de SW4 – Sem função;
- Pressionamento longo de SW1 – Volta para o modo de configuração;
- Pressionamento longo de SW2 – Sem função;
- Pressionamento longo de SW3 – Sem função;
- Pressionamento longo de SW4 – Sem função;.

2.5 TESTES DE DESEMPENHO DA REDE

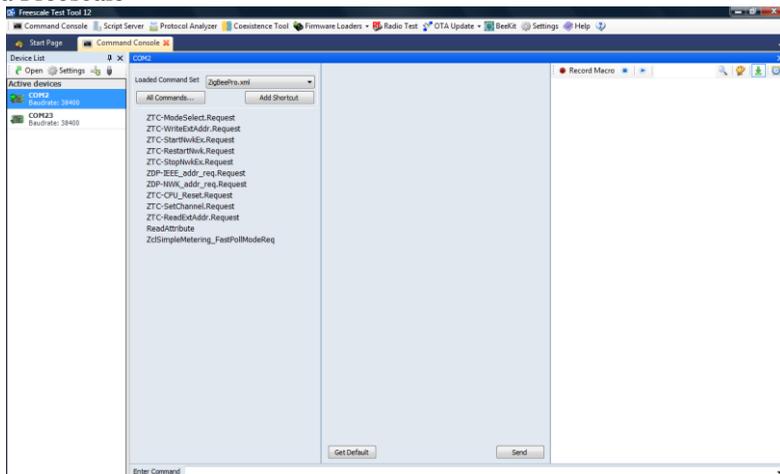
2.5.1 Teste de validação da comunicação dos dispositivos via protocolo Zigbee

O objetivo deste teste é avaliar a rede *Zigbee*, quanto a sua inicialização, atribuição de endereços aos dispositivos, associações entre

dispositivos e troca de mensagens.

A metodologia utilizada para a execução do teste compreende iniciar a rede conforme exposto na seção 2.4.2, através do *software* de desenvolvimento “Test Tool 12” disponibilizado pela *Freescale*. Este *software* funciona como um *sniffer*, monitorando os comandos da rede através dos comandos do ZTC e sua interface com o usuário é mostrada na Figura 20.

Figura 20: Interface do “Command Console” da ferramenta “Test Tool” da Freescale



Fonte: Produção do próprio autor.

Para este teste foram utilizados dois dispositivos, um coordenador da rede e um *end device*. Ambos os dispositivos foram conectados ao computador via USB, conforme mostra a Figura 21, que documenta o *setup* do teste. Os dois dispositivos estavam gravados com um *software* onde a comunicação via ZTC está habilitada.

Figura 21: Setup de teste de validação da comunicação *Zigbee*



Fonte: Produção do próprio autor.

Caso aconteça algum evento ou comando na execução da rede *Zigbee*, este evento ou comando será registrado pelo *software Test Tool*, documentando de forma clara o que aconteceu na operação da rede *Zigbee*.

Os resultados do teste são mostrados abaixo:

1. Botão SW1 pressionado no dispositivo coordenador da rede: Inicia a rede.

Leitura realizada através dos *software Test Tool* para o coordenador da rede:

```
[16:41:22.0272] RX: ZDO_EventOccurred.Indication 02 A0
E6 01 03 44
Sync          [1 byte ] = 02
OpGroup      [1 byte ] = A0
OpCode       [1 byte ] = E6
Length       [1 byte ] = 01
EventStatus  [1      byte   ] =      03
(DeviceinCoordinatorstartingstate)
```

CRC [1 byte] = 44

```
[16:41:22.0272] RX: NLME-NETWORK-FORMATION.Request 02 96
35 0A 00 00 04 03 0F 00 AA 1A 00 11
Sync [1 byte ] = 02
OpGroup [1 byte ] = 96
OpCode [1 byte ] = 35
Length [1 byte ] = 0A
ScanChannels [4 bytes] = 04 00 00 00
ScanDuration [1 byte ] = 03
BeaconOrder [1 byte ] = 0F
SuperframeOrder [1 byte ] = 00
PANId [2 bytes] = 1A AA
BatteryLifeExtension [1 byte ] = 00 (false)
CRC [1 byte ] = 11
```

```
[16:41:22.0555] RX: NLME-NETWORK-FORMATION.Confirm 02 97
42 01 00 D4
Sync [1 byte ] = 02
OpGroup [1 byte ] = 97
OpCode [1 byte ] = 42
Length [1 byte ] = 01
Status [1 byte ] = 00 (SUCCESS)
CRC [1 byte ] = D4
```

```
[16:41:22.0561] RX: ZDO_EventOccurred.Indication 02 A0
E6 01 10 57
Sync [1 byte ] = 02
OpGroup [1 byte ] = A0
OpCode [1 byte ] = E6
Length [1 byte ] = 01
EventStatus [1 byte ] = 10
(DeviceinCoordinatorRunningstate)
CRC [1 byte ] = 57
```

O primeiro comando registrado foi o “RX: ZDO_EventOccurred.Indication”, onde o Test Tool registrou o evento de que o botão foi pressionado, e que o coordenador entrou no estado “DeviceinCoordinatorstartingstate”.

No segundo comando “RX: NLME-NETWORK-FORMATION.Request”, registrou-se que o processo de inicialização da rede começou. O terceiro comando “RX: NLME-NETWORK-FORMATION.Confirm” indica a confirmação de que o processo de inicialização foi realizado com sucesso.

No último comando registrado “RX: ZDO_EventOccurred.Indication”, verifica-se que o dispositivo

coordenador da rede passou para o próximo estado de operação “DeviceinCoordinatorRunningstate”.

Verifica-se neste log de mensagens que o endereço de rede, também chamado de PAN ID é 1AAA.

2. Botão SW1 pressionado no dispositivo *end device* da rede: Dispositivo procura uma rede válida e pede para se juntar a esta rede.

Leitura realizada através dos *software Test Tool* para o coordenador da rede:

```
RX: NLME-JOIN.Indication 02 97 45 0D 70 79 01 A4 07 44
0F C2 50 00 8C 00 00 21
Sync                [1 byte ] = 02
OpGroup             [1 byte ] = 97
OpCode              [1 byte ] = 45
Length              [1 byte ] = 0D
ShortAddress        [2 bytes] = 79 70
ExtendedAddress     [8 bytes] = 00 50 C2 0F 44 07 A4
01
CapabilityInformation [1 byte ] = 8C
Rejoin              [1 byte ] = 00
SecureJoin          [1 byte ] = 00 (false)
CRC                 [1 byte ] = 21
```

Com isso confirma-se que o dispositivo sensor se conectou a rede com sucesso. A ele foi atribuído o endereço de rede 00 50 C2 0F 44 07 A4 01.

Leitura realizada através dos *software Test Tool* para o dispositivo configurado como *end device*:

```
RX: ZDO_EventOccurred.Indication 02 A0 E6 01 00 47
Sync                [1 byte ] = 02
OpGroup             [1 byte ] = A0
OpCode              [1 byte ] = E6
Length              [1 byte ] = 01
EventStatus         [1      byte   ] =      01
(DeviceinNetworkDiscoveryState)
CRC                 [1 byte ] = 47
RX: ZDO_EventOccurred.Indication 02 A0 E6 01 01 46
Sync                [1 byte ] = 02
OpGroup             [1 byte ] = A0
```

```

OpCode      [1 byte ] = E6
Length      [1 byte ] = 01
EventStatus [1      byte   ] =      01
(DeviceInNetworkDiscoveryState)
CRC         [1 byte ] = 46
RX: NLME-NETWORK-DISCOVERY.Request 02 96 34 05 00 40 00
00 03 E4
Sync        [1 byte ] = 02
OpGroup     [1 byte ] = 96
OpCode      [1 byte ] = 34
Length      [1 byte ] = 05
ScanChannels [4 bytes] = 00 00 40 00
ScanDuration [1 byte ] = 03
CRC         [1 byte ] = E4
RX: NLME-NETWORK-DISCOVERY.Confirm 02 97 41 12 01 AA 1A
0E B0 0E 24 13 C2 50 00 0E 01 02 0F 0F 01 00 6C
Sync        [1 byte ] = 02
OpGroup     [1 byte ] = 97
OpCode      [1 byte ] = 41
Length      [1 byte ] = 12
NetworkCount [1 byte ] = 01
NetworkDescriptor [16 bytes] = AA 1A 0E B0 0E 24 13 C2
50 00 0E 01 02 0F 0F 01
Status      [1 byte ] = 00 (SUCCESS)
CRC         [1 byte ] = 6C
RX: NLME-NETWORK-DISCOVERY.Request 02 96 34 05 00 40 00
00 03 E4
Sync        [1 byte ] = 02
OpGroup     [1 byte ] = 96
OpCode      [1 byte ] = 34
Length      [1 byte ] = 05
ScanChannels [4 bytes] = 00 00 40 00
ScanDuration [1 byte ] = 03
CRC         [1 byte ] = E4
RX: NLME-NETWORK-DISCOVERY.Confirm 02 97 41 12 01 AA 1A
0E B0 0E 24 13 C2 50 00 0E 01 02 0F 0F 01 00 6C
Sync        [1 byte ] = 02
OpGroup     [1 byte ] = 97
OpCode      [1 byte ] = 41
Length      [1 byte ] = 12
NetworkCount [1 byte ] = 01
NetworkDescriptor [16 bytes] = AA 1A 0E B0 0E 24 13 C2
50 00 0E 01 02 0F 0F 01
Status      [1 byte ] = 00 (SUCCESS)
CRC         [1 byte ] = 6C
RX: NLME-NETWORK-DISCOVERY.Request 02 96 34 05 00 40 00
00 03 E4
Sync        [1 byte ] = 02

```

```

OpGroup      [1 byte ] = 96
OpCode       [1 byte ] = 34
Length       [1 byte ] = 05
ScanChannels [4 bytes] = 00 00 40 00
ScanDuration [1 byte ] = 03
CRC          [1 byte ] = E4
RX: NLME-NETWORK-DISCOVERY.Confirm 02 97 41 12 01 AA 1A
0E B0 0E 24 13 C2 50 00 0E 01 02 0F 0F 01 00 6C
Sync         [1 byte ] = 02
OpGroup      [1 byte ] = 97
OpCode       [1 byte ] = 41
Length       [1 byte ] = 12
NetworkCount [1 byte ] = 01
NetworkDescriptor [16 bytes] = AA 1A 0E B0 0E 24 13 C2
50 00 0E 01 02 0F 0F 01
Status       [1 byte ] = 00 (SUCCESS)
CRC          [1 byte ] = 6C
RX: ZDO_EventOccurred.Indication 02 A0 E6 01 02 45
Sync         [1 byte ] = 02
OpGroup      [1 byte ] = A0
OpCode       [1 byte ] = E6
Length       [1 byte ] = 01
EventStatus  [1 byte ] = 02 (DeviceJoinNetworkstate)
CRC          [1 byte ] = 45
RX: NLME-JOIN.Request 02 96 38 12 00 00 00 00 00 00
00 00 00 00 40 00 00 03 00 01 00 FE
Sync         [1 byte ] = 02
OpGroup      [1 byte ] = 96
OpCode       [1 byte ] = 38
Length       [1 byte ] = 12
ExtPANId     [8 bytes] = 00 00 00 00 00 00 00 00
JoinAsRouter [1 byte ] = 00 (false)
RejoinNetwork [1 byte ] = 00
ScanChannels [4 bytes] = 00 00 40 00
ScanDuration [1 byte ] = 03
PowerSource  [1 byte ] = 00
RxOnWhenIdle [1 byte ] = 01 (true)
SecurityEnable [1 byte ] = 00
CRC          [1 byte ] = FE
RX: NLME-JOIN.Confirm 02 97 46 06 6F 79 AA 1A 00 00 71
Sync         [1 byte ] = 02
OpGroup      [1 byte ] = 97
OpCode       [1 byte ] = 46
Length       [1 byte ] = 06
ShortAddress [2 bytes] = 79 6F
PANId        [2 bytes] = 1A AA
HaveNetworkKey [1 byte ] = 00 (false)
Status       [1 byte ] = 00 (SUCCESS)

```

```

CRC                [1 byte ] = 71
RX: ZDO_EventOccurred.Indication 02 A0 E6 01 05 42
Sync               [1 byte ] = 02
OpGroup            [1 byte ] = A0
OpCode             [1 byte ] = E6
Length             [1 byte ] = 01
EventStatus        [1      byte      ] =      05
(DeviceinEndDeviceRunningstate)
CRC                [1 byte ] = 42

```

Verifica-se analisando as mensagens acima, que o dispositivo procurou uma rede válida para poder se conectar e ao encontrar fez o pedido de conexão para o coordenador da rede. Em seguida recebeu uma mensagem indicando que a conexão da rede foi realizada com sucesso e mudou seu estado para em funcionamento.

3. Para avaliar a troca de dados entre o dispositivo coordenador e o dispositivo sensor conectado a rede, foi enviado um comando através do *software Test Tool* para fazer com o coordenador da rede peça o endereço de rede para o dispositivo sensor (ZDP-IEEE_addr_req.Request) e também um comando pedindo alguns de seus atributos (ZDP-NodeDescriptor.Request).

Leitura realizada através dos *software Test Tool* para o comando ZDP-IEEE_addr_req.Request:

```

TX: ZDP-IEEE_addr_req.Request 02 A2 01 06 6F 79 6F 79 01
00 A4
Sync               [1 byte ] = 02
OpGroup            [1 byte ] = A2
OpCode             [1 byte ] = 01
Length             [1 byte ] = 06
DestAddress        [2 bytes] = 79 6F
NWKAddress         [2 bytes] = 79 6F
RequestType        [1 byte ] = 01
StartIndex         [1 byte ] = 00
CRC                [1 byte ] = A4

RX: APSDE-DATA.Request 02 9C 00 17 02 6F 79 00 00 00 00
00 00 00 00 01 00 00 05 04 6F 79 01 00 00 0A 82
Sync               [1 byte ] = 02
OpGroup            [1 byte ] = 9C
OpCode             [1 byte ] = 00
Length             [1 byte ] = 17
DstAddrMode        [1 byte ] = 02

```

```

DstAddress      [8 bytes] = 00 00 00 00 00 00 79 6F
DstEndpoint     [1 byte ] = 00
ProfileId       [2 bytes] = 00 00
ClusterId       [2 bytes] = 00 01
SrcEndpoint     [1 byte ] = 00
asduLength      [1 byte ] = 05
Asdu            [5 bytes] = 04 6F 79 01 00
TxOptions       [1 byte ] = 00
RadiusCounter   [1 byte ] = 0A
CRC             [1 byte ] = 82

```

```

RX: APSDE-DATA.Confirm 02 9D 00 0D 02 6F 79 00 00 00 00
00 00 00 00 00 86 02

```

```

Sync           [1 byte ] = 02
OpGroup        [1 byte ] = 9D
OpCode         [1 byte ] = 00
Length         [1 byte ] = 0D
DstAddrMode    [1 byte ] = 02
DstAddress     [8 bytes] = 00 00 00 00 00 00 79 6F
DstEndpoint    [1 byte ] = 00
SrcEndpoint    [1 byte ] = 00
Status         [1 byte ] = 00 (gSuccess)
TxTime         [4 bytes] = 86 00 00 00
ConfirmID      [1 byte ] = 00
CRC            [1 byte ] = 02

```

```

RX: APSDE-DATA.Indication 02 9D 01 1D 02 00 00 00 02 6F
79 00 00 00 01 80 0D 04 00 01 40 06 54 03 C2 50 00 6F 79
00 00 00 D0 5B

```

```

Sync           [1 byte ] = 02
OpGroup        [1 byte ] = 9D
OpCode         [1 byte ] = 01
Length         [1 byte ] = 1D
DestAddrMode   [1 byte ] = 02 (16bitAddrandDstEndpoint)
DstAddress     [2 bytes] = 00 00
DstEndpoint    [1 byte ] = 00
SrcAddrMode    [1 byte ] = 02
SrcAddress     [2 bytes] = 79 6F
SrcEndpoint    [1 byte ] = 00
ProfileId      [2 bytes] = 00 00
ClusterId      [2 bytes] = 80 01
asduLength     [1 byte ] = 0D
asdu           [13 bytes] = 04 00 01 40 06 54 03 C2 50
00 6F 79 00
Status         [1 byte ] = 00 (Success)
WasBroadcast   [1 byte ] = 00 (FALSE)
SecurityStatus [1 byte ] = D0
LinkQuality    [1 byte ] = 00

```

```

RxTime          [4 bytes] = 00 00 00 00
iMsgType        [1 byte ] = 00
pNext           [2 bytes] = 00 00
iDataSize       [1 byte ] = 00
pData           [2 bytes] = 00 00
iBufferNumber   [1 byte ] = 00
CRC             [1 byte ] = 5B

```

```

RX: ZDP-IEEE_addr.response 02 A0 81 0D 00 01 40 06 54 03
C2 50 00 6F 79 00 00 B8

```

```

Sync            [1 byte ] = 02
OpGroup         [1 byte ] = A0
OpCode          [1 byte ] = 81
Length          [1 byte ] = 0D
Status          [1 byte ] = 00 (Success)
IEEEAddrRemoteDev [8 bytes] = 00 50 C2 03 54 06 40
01
NWKAddrRemoteDev [2 bytes] = 79 6F
NumOfAssociatedDevice [1 byte ] = 00
StartIndex      [1 byte ] = 00
CRC             [1 byte ] = B8

```

Leitura realizada através dos *software Test Tool* para o comando ZDP-NodeDescriptor.Request:

```

TX: ZDP-NodeDescriptor.Request 02 A2 02 04 6F 79 6F 79
A4

```

```

Sync            [1 byte ] = 02
OpGroup         [1 byte ] = A2
OpCode          [1 byte ] = 02
Length          [1 byte ] = 04
DestAddress     [2 bytes] = 79 6F
NwkAddressOfInterest [2 bytes] = 79 6F
CRC             [1 byte ] = A4

```

```

RX: APSDE-DATA.Request 02 9C 00 15 02 6F 79 00 00 00 00
00 00 00 00 02 00 00 03 05 6F 79 00 0A 85

```

```

Sync            [1 byte ] = 02
OpGroup         [1 byte ] = 9C
OpCode          [1 byte ] = 00
Length          [1 byte ] = 15
DstAddrMode     [1 byte ] = 02
DstAddress      [8 bytes] = 00 00 00 00 00 00 79 6F
DstEndpoint     [1 byte ] = 00
ProfileId       [2 bytes] = 00 00
ClusterId       [2 bytes] = 00 02

```

```

SrcEndpoint    [1 byte ] = 00
asduLength    [1 byte ] = 03
Asdu          [3 bytes] = 05 6F 79
TxOptions     [1 byte ] = 00
RadiusCounter [1 byte ] = 0A
CRC           [1 byte ] = 85

```

```

RX: APSDE-DATA.Confirm 02 9D 00 0D 02 6F 79 00 00 00 00
00 00 00 00 00 87 03

```

```

Sync          [1 byte ] = 02
OpGroup       [1 byte ] = 9D
OpCode        [1 byte ] = 00
Length        [1 byte ] = 0D
DstAddrMode   [1 byte ] = 02
DstAddress    [8 bytes] = 00 00 00 00 00 00 79 6F
DstEndpoint   [1 byte ] = 00
SrcEndpoint   [1 byte ] = 00
Status        [1 byte ] = 00 (gSuccess)
TxTime        [4 bytes] = 87 00 00 00
ConfirmID     [1 byte ] = 00
CRC           [1 byte ] = 03

```

```

RX: APSDE-DATA.Indication 02 9D 01 1E 02 00 00 00 02 6F
79 00 00 00 02 80 0E 05 00 6F 79 02 40 8C 10 02 50 00 00
00 00 00 00 D1 56

```

```

Sync          [1 byte ] = 02
OpGroup       [1 byte ] = 9D
OpCode        [1 byte ] = 01
Length        [1 byte ] = 1E
DestAddrMode  [1 byte ] = 02 (16bitAddrandDstEndpoint)
DstAddress    [2 bytes] = 00 00
DstEndpoint   [1 byte ] = 00
SrcAddrMode   [1 byte ] = 02
SrcAddress     [2 bytes] = 79 6F
SrcEndpoint   [1 byte ] = 00
ProfileId     [2 bytes] = 00 00
ClusterId     [2 bytes] = 80 02
asduLength    [1 byte ] = 0E
asdu          [14 bytes] = 05 00 6F 79 02 40 8C 10 02
50 00 00 00 00
Status        [1 byte ] = 00 (Success)
WasBroadcast  [1 byte ] = 00 (FALSE)
SecurityStatus [1 byte ] = D1
LinkQuality   [1 byte ] = 00
RxTime        [4 bytes] = 00 00 00 00
iMsgType      [1 byte ] = 00
pNext         [2 bytes] = 00 00
iDataSize     [1 byte ] = 00

```

```

pData          [2 bytes] = 00 00
iBufferNumber [1 byte ] = 00
CRC            [1 byte ] = 56

RX: ZDP-NodeDescriptor.Response 02 A0 82 0D 00 6F 79 02
40 8C 10 02 50 00 00 00 00 B5
Sync          [1 byte ] = 02
OpGroup       [1 byte ] = A0
OpCode        [1 byte ] = 82
Length        [1 byte ] = 0D
Status        [1 byte ] = 00
NWKAddressOfInterest [2 bytes] = 79 6F
NODE-DESCRIPTOR_FIRST_BYTE [1 byte ] = 02
NODE-DESCRIPTOR_SECOND_BYTE [1 byte ] = 40
NODE-DESCRIPTOR_MACcapabilityFlags_BYTE [1 byte ]
= 8C
NODE-DESCRIPTOR_ManufacturerCode_BYTES [2 bytes]
= 02 10
NODE-DESCRIPTOR_MaximumBufferSize_BYTE [1 byte ]
= 50
NODE-DESCRIPTOR_MaximumTransferSize_BYTES [2 bytes]
= 00 00
NODE-DESCRIPTOR_ServerMask_BYTES [2 bytes]
= 00 00
NODE-DESCRIPTOR_MaximumOutTransferSize_BYTES [2 bytes]
= 00 00
NODE-DESCRIPTOR_CapabilityField_BYTES [1 byte ]
= 00
CRC            [1 byte ] = B5

```

Ao final dos testes de inicialização da rede, foi verificado que o dispositivo coordenador da rede está formando a rede adequadamente, conforme o esperado, e os demais dispositivos da rede se conectaram ao coordenador da forma correta.

Observou-se que a troca de mensagens foi bem sucedida e que os dispositivos da rede estão se comunicando conforme o esperado. Com isso esta etapa de inicialização da rede e conexão de dispositivos está validada e funcionando corretamente.

2.5.2 Teste de medição do alcance da rede

O objetivo deste teste é avaliar o alcance do sistema de aquisição de dados implementado através do protocolo de comunicação sem fios *Zigbee*. Para a realização deste teste será monitorado a distância física de alcance para a troca de mensagens da rede e a

qualidade do sinal transmitido através do parâmetro *Link Quality Indicator* (LQI), explicado no Apêndice A.

Este teste foi realizado em três ambientes diferentes, com o objetivo de verificar o desempenho da rede em cada um deles. Os ambientes são descritos conforme apontado abaixo:

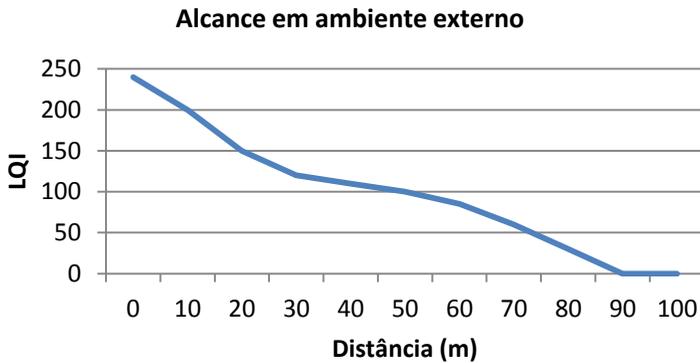
- 1) Ambiente externo aberto: Local onde não há obstáculos para os sinais de radiofrequência da rede entre o dispositivo coordenador e o nó final.
- 2) Ambiente interno sem paredes: Local onde há poucos obstáculos para os sinais de radiofrequência da rede.
- 3) Ambiente interno com paredes: Local onde a rede provavelmente deve ser utilizada, e que pode apresentar restrições para o alcance e qualidade dos sinais da rede.

O protocolo *Zigbee* tem em sua camada de rede uma variável que é monitorada para indicar a qualidade do *link* da rede. O nome desta variável é *Link Quality Indication* – LQI, a qual será utilizada como referência para as medições de qualidade do sinal da rede. Este sinal varia em uma faixa hexadecimal de 0x00 a 0xFF (0 a 255 em decimal), onde quanto menor o valor pior é a qualidade do sinal da rede. Considerou-se neste trabalho um valor mínimo de LQI igual a 50. O LQI entre os nós da rede será monitorado através do programa “Freescale Range Test” da Freescale. A distância alcançada pelo sistema será medida em metros.

Resultados

- 1) Ambiente externo aberto:

Figura 22: Gráfico de medição do alcance da rede em ambiente externo

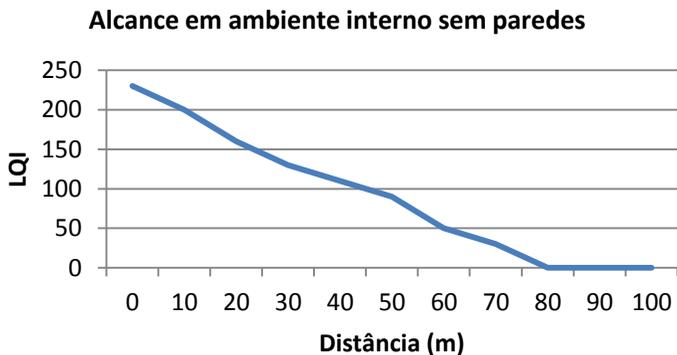


Fonte: Produção do próprio autor.

Em ambiente externo aberto, a distância máxima alcançada com a rede *Zigbee* ficou em aproximadamente 80m.

2) Ambiente interno sem paredes:

Figura 23: Gráfico de medição do alcance da rede em ambiente interno sem paredes

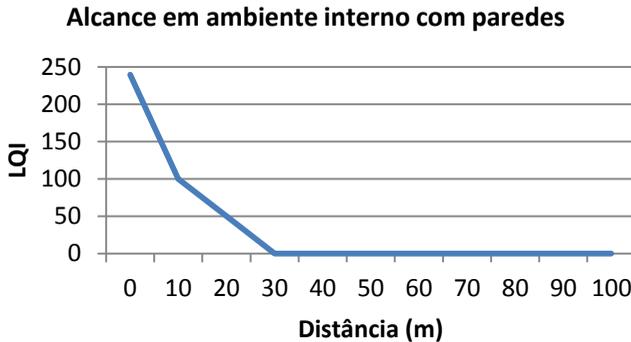


Fonte: Produção do próprio autor.

Em ambiente interno sem paredes, a distância máxima alcançada com a rede *Zigbee* ficou em aproximadamente 70m.

3) Ambiente interno com paredes:

Figura 24: Gráfico de medição do alcance da rede em ambiente interno com paredes



Fonte: Produção do próprio autor.

Em ambiente interno com paredes, a distância máxima alcançada com a rede *Zigbee* ficou em aproximadamente 13 metros.

Após os testes, foi possível verificar que o sistema de transmissão de dados sem fio *Zigbee* possibilita a comunicação entre os dispositivos conforme especificado no protocolo, atingindo um alcance próximo a 80 metros em lugares abertos, sendo que o protocolo *Zigbee* estabelece um parâmetro de 100 metros nesta condição e atingindo um alcance próximo de 13 metros em ambiente fechado, com o protocolo estabelecendo uma distância de até 30 metros dependendo do número de obstáculos.

CAPÍTULO 3 – SOFTWARE DE INTERFACE COM O USUÁRIO

A interface com o usuário desenvolvida para este projeto tem como objetivo proporcionar ao usuário do sistema de aquisição de dados utilizando a rede sem fio *Zigbee*, uma forma fácil de inicializar e controlar a rede, identificar e visualizar os dispositivos presentes na sua rede, além de facilmente visualizar e manipular os dados adquiridos através dos dispositivos dedicados à aquisição dos sinais bioelétricos.

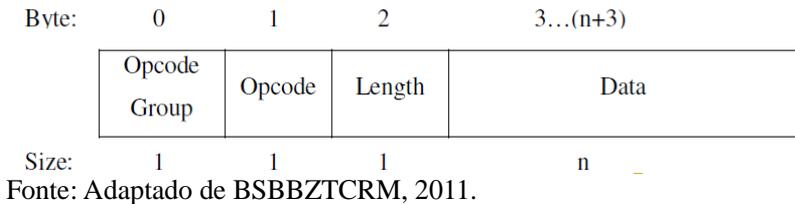
Esta interface com o usuário foi construída utilizando o ambiente de desenvolvimento LabVIEW, fabricado pela empresa *National Instruments* (NI). Esta é uma ferramenta otimizada para aplicações de engenharia, mais especificamente para sistemas de teste, controle e monitoramento. Através do LabVIEW, é possível fazer a aquisição de sinais, fazer a análise dos dados e a comunicação ou armazenamento dos resultados de forma rápida e fácil. O LabVIEW utiliza um ambiente gráfico de programação por fluxo de dados, em substituição às linguagens convencionais baseadas em texto. A versão utilizada para o desenvolvimento deste trabalho é o LabVIEW 2009.

3.1 COMUNICAÇÃO ENTRE O SOFTWARE DE INTERFACE COM USUÁRIO E O FIRMWARE ZIGBEE

A interface com o usuário se comunica com o dispositivo coordenador da rede, via comunicação USB, e é através desta comunicação que todos os dados mostrados para o usuário são captados da rede *Zigbee*.

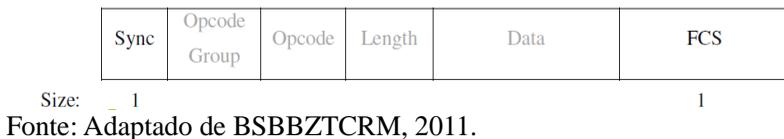
Conforme apresentado no capítulo 2, a comunicação entre a interface com o usuário e a rede *Zigbee* é feita através da estrutura de dados chamada de ZTC. Esta estrutura tem como padrão de comunicação o formato de mensagem mostrado na Figura 25. Nela tem-se um identificador chamado *Opcode Group*, o qual é o byte 0 da mensagem e delimita o grupo de mensagens que será trocado. O segundo *byte*, chamado de *Opcode* é quem identifica qual a mensagem deve ser enviada ou, esta sendo recebida dentro do grupo de mensagens definido pelo *Opcode Group*. O terceiro *byte* determina o tamanho da mensagem, visto que a mesma pode variar de acordo com o tamanho do pacote de dados a ser traçado. Do quarto até o último *byte* da mensagem tem-se o pacote de dados a ser transmitido.

Figura 25: Formato de mensagem do ZTC



Para identificar o início de cada mensagem e para garantir sua integridade, o ZTC utiliza um *byte* a mais no início da mensagem e um *byte* adicional ao final da mensagem. A Figura 26 mostra o formato genérico da mensagem, onde o primeiro *byte* é o *byte* de sincronismo e o último *byte* é o *byte* chamado de *checksum*, e que mostra a soma dos *bytes* anteriores.

Figura 26: Formato genérico de mensagem do ZTC



As tabelas no Anexo A mostram alguns dos comandos do ZTC utilizados pela comunicação implementada no *software* do Labview.

3.2 FUNCIONAMENTO DO SOFTWARE DE INTERFACE COM O USUÁRIO

O *software* é composto pela tela de configuração e por telas específicas que mostram o estado da rede e os dispositivos disponíveis para a aquisição de dados. Pode-se dividir o *software* em forma de blocos conforme mostrado na Figura 27.

Figura 27: Diagrama de blocos do *software* de interface com o usuário



Fonte: Produção do próprio autor.

Nesta figura podemos perceber que temos um bloco dedicado ao envio e recepção de mensagens para o dispositivo coordenador da rede. Outro bloco que é a máquina de estados que garante o funcionamento do *software*, a qual inicializa a rede, e gerencia o processo de detecção de dispositivos e atributos, além da aquisição periódica dos dados dos dispositivos sensores.

Outro bloco é dedicado ao gerenciamento de dados da rede, além de organização e disponibilização de informações da rede *Zigbee* para o usuário através da tela “Ver Estado da Rede”. Temos ainda uma estrutura que monitora o estado de cada botão da interface com o usuário. Esta estrutura toma as ações adequadas para cada função disponível por estes botões. E por último há a estrutura de gravação de dados, a qual salva os dados adquiridos em arquivo texto caso o usuário deseje.

A tela inicial é a tela de configuração, mostrada na Figura 28.

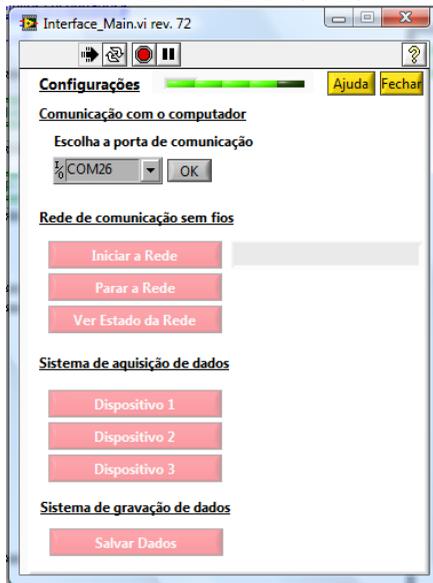
Esta tela é composta por:

- Barra de *status*: Atualiza periodicamente para mostrar ao usuário que o *software* está funcionando.
- Botão de Ajuda: Caso o usuário tenha alguma dúvida no funcionamento quanto ao sistema de aquisição de dados pode procurar neste botão.
- Botão Fechar: Encerra o funcionamento do *software*.
- Barra para escolha da porta de comunicação: É onde o usuário

configura a porta de comunicação a ser utilizada para conexão do dispositivo coordenador da rede.

- Botão OK: Botão para confirma a escolha da porta de comunicação.
- Botão Iniciar a Rede: Inicia a busca de dispositivos na rede.
- Botão Parar a Rede: Para a rede
- Botão Ver Estado da Rede: Abre a janela “Ver estado da Rede” e mostra os atributos dos dispositivos conectados na rede.
- Botão Dispositivo 1: Abre a janela que mostra os dispositivos disponíveis para aquisição de dados no dispositivo 1.
- Botão Dispositivo 2: Abre a janela que mostra os dispositivos disponíveis para aquisição de dados no dispositivo 2.
- Botão Dispositivo 3: Abre a janela que mostra os dispositivos disponíveis para aquisição de dados no dispositivo 3.

Figura 28: Tela inicial da interface com o usuário



Fonte: Produção do próprio autor.

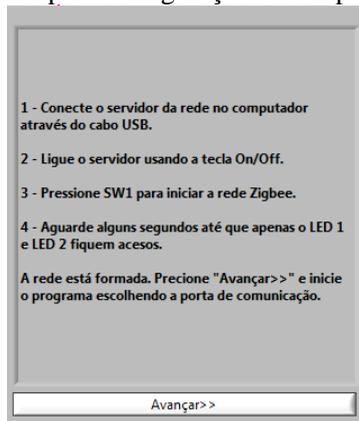
Para iniciar o uso do sistema, é necessário:

- 1 - Conectar o dispositivo coordenador da rede no computador através do cabo USB.
- 2 - Ligar o coordenador usando a tecla On/Off.

- 3 - Pressionar SW1 no coordenador para iniciar a rede *Zigbee*.
- 4 - Aguardar alguns segundos enquanto os LEDs 1, 2, 3 e 4 piscam até que apenas o LED 1 e LED 2 fiquem acesos.

A Figura 29 é um aviso para o usuário do sistema fazer a inicialização dos dispositivos ao iniciar o *software*. Esta é uma mensagem orientativa e para continuar a utilização do sistema, o usuário precisa pressionar o botão avançar.

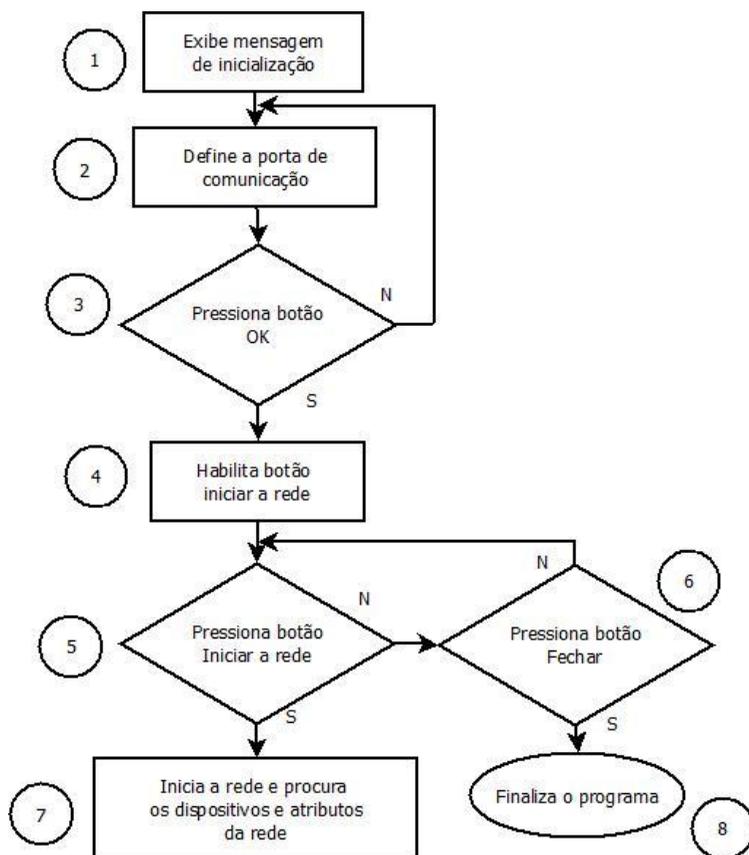
Figura 29: Aviso inicial para configuração dos dispositivos



Fonte: Produção do próprio autor.

Até que a porta de comunicação seja escolhida os botões ficam desabilitados e a rede não pode ser iniciada. O fluxograma mostrado na Figura 30 e a Tabela 12 mostram as etapas iniciais utilizadas para iniciar o sistema de aquisição de dados.

Figura 30: Fluxograma da tela inicial da interface com o usuário



Fonte: Produção do próprio autor.

Tabela 12: Descrição das etapas do fluxograma da tela inicial da interface com o usuário

Etapa	Descrição
1	Ao iniciar o programa uma mensagem é mostrada para o usuário para que o mesmo verifique se os dispositivos foram iniciados corretamente.
2	O usuário define a porta de comunicação a ser utilizada pelo

	sistema.
3	Usuário confirma a porta de comunicação escolhida.
4	Botão iniciar a rede é habilitado e o usuário pode então iniciar o uso do sistema.
5	<i>Software</i> aguarda que o usuário pressione o botão iniciar a rede para procurar pelos dispositivos ou que finalize o programa.
6	Verifica se o usuário pressionou o botão para fechar o programa.
7	Se o botão iniciar a rede foi pressionado, procura os dispositivos presentes na rede e seus atributos.
8	Finaliza a execução do programa.

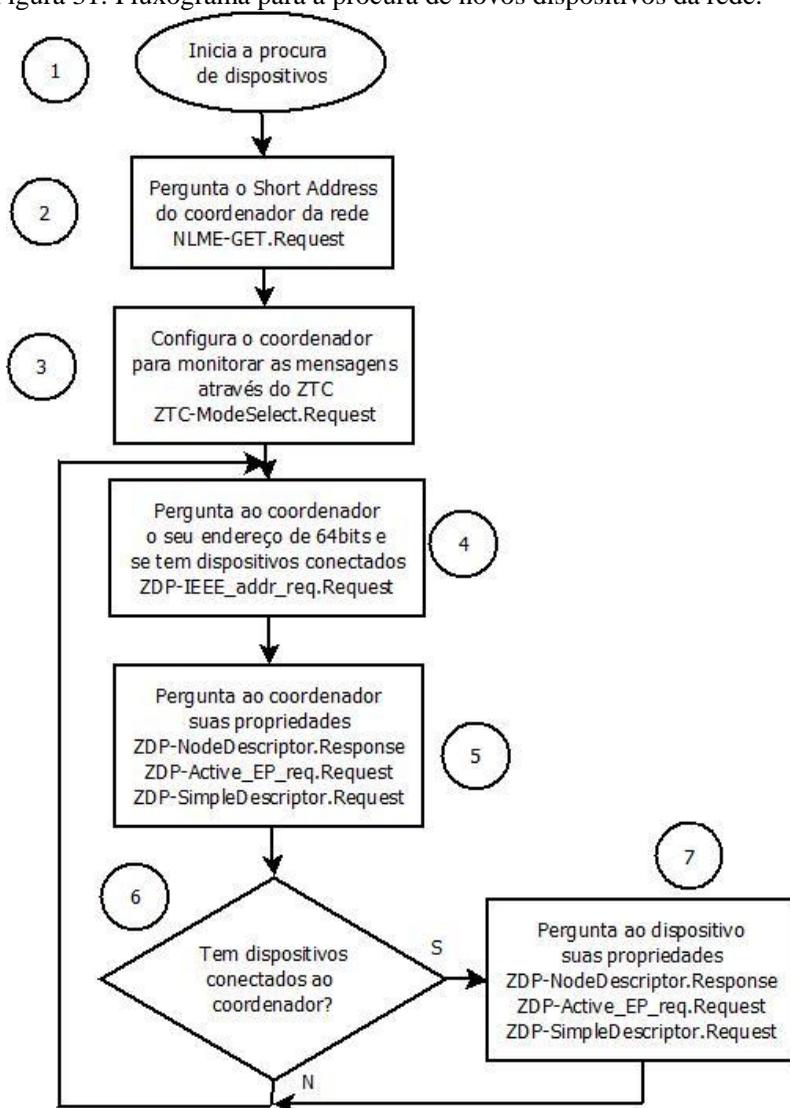
Fonte: Produção do próprio autor.

Para inicializar os dispositivos sensores da rede, deve-se:

- 1 - Ligar o dispositivo sensor (bateria ou na rede).
- 2 - Ligar o dispositivo sensor usando a tecla On/Off.
- 3 - Pressionar SW1 no dispositivo sensor para que ele procure uma rede e se conecte a ela.
- 4 - Aguardar alguns segundos enquanto os LEDs 1, 2, 3 e 4 piscam até que apenas o LED 1 fique aceso.

A lógica utilizada pelo *software* de interface com o usuário para a procura de dispositivos e atributos dos dispositivos encontrados é descrita no fluxograma da Figura 31 e na Tabela 13.

Figura 31: Fluxograma para a procura de novos dispositivos da rede.



Fonte: Produção do próprio autor.

Tabela 13: Descrição das etapas do fluxograma para a procura de novos dispositivos da rede

Etapa	Descrição
1	Inicia a procura de dispositivos após o botão “iniciar a rede” ser pressionado.
2	Pergunta ao dispositivo coordenador da rede o seu endereço de rede “ <i>short address</i> ” através do comando NLME-GET.Request.
3	Configura o coordenador da rede para o monitoramento de mensagens via ZTC através do comando ZTC-ModeSelect.Request.
4	Pergunta ao dispositivo coordenador da rede o seu endereço de 64bits e verifica se existem outros dispositivos conectados ao coordenador da rede através do comando ZDP-IEEE_addr_req.Request.
5	Pergunta ao dispositivo coordenador da rede suas propriedades através dos comandos ZDP-NodeDescriptor.Request, ZDP-Active_EP_req.Request e ZDP-SimpleDescriptor.Request.
6	Analisa se existem dispositivos sensores conectados a rede e caso sim, pergunta seu endereço de rede e suas propriedades. Caso não existam outros dispositivos além do coordenador, retorna para a etapa 4, para confirmar se o coordenador da rede ainda está ativo.
7	Pergunta aos dispositivos sensores suas propriedades através dos comandos ZDP-IEEE_addr_req.Request, ZDP-NodeDescriptor.Request, ZDP-Active_EP_req.Request e ZDP-SimpleDescriptor.Request.

Fonte: Produção do próprio autor.

As propriedades dos dispositivos são armazenadas em uma tabela para que o *software* possa gerenciar corretamente o funcionamento da troca de mensagens. Estas propriedades são mostradas para o usuário da rede ao se pressionar o botão “Ver estado da Rede”, na tela inicial da interface com o usuário. Ao se pressionar este botão uma nova janela é aberta, conforme mostrado na Figura 32.

Figura 32: Tabela com as propriedades dos dispositivos da rede.

	Coordenador	Dispositivo 1	Dispositivo 2	Dispositivo 3
Network short address	0000	796F		
IEEE Addr 64Bits	0050 C209 6810 8C0F			
Router				
Logical Type	436F 6F72 6465			
APS	40			
MAC	8E			
MFCCode				
Buffer				
Tx				
EP	08			
ID				
Devices				
Ver				
Flags				

Fonte: Produção do próprio autor.

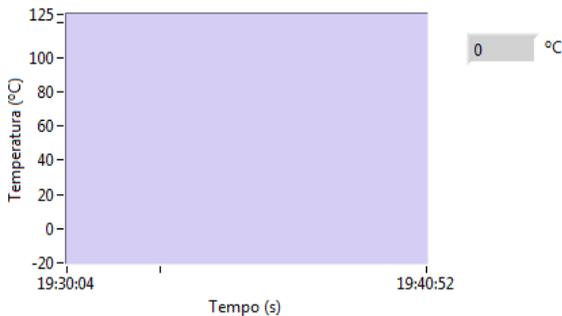
São várias as propriedades necessárias para um bom funcionamento da rede e para que seja possível se conhecer qual aplicativo que cada dispositivo tem para contribuir com a rede. Entre eles, podemos destacar, o endereço de rede de cada dispositivo colocado na Figura 32, como “*Network short address*“, o endereço de cada dispositivo chamado de “*IEEE Addr 64Bits*“, o tipo lógico do dispositivo, se ele é coordenador, *router* ou *end device*. Além de informação referente à camada de aplicação, e camada MAC.

Após a inicialização da rede e captura dos parâmetros de cada dispositivo, o *software* habilitará os botões para acesso as aquisições de cada dispositivo sensor da rede. Dependendo no número de dispositivos disponíveis na rede os botões “Dispositivo 1”, “Dispositivo 2” e “Dispositivo 3” serão habilitados para serem pressionados.

Ao pressionar um destes botões uma nova tela irá se abrir e o usuário do sistema terá acesso aos dados que estão sendo adquiridos por aquele dispositivo. A Figura 33 mostra a tela de monitoramento de aquisição dos dados de um dispositivo. Nesta tela o usuário poderá acompanhar em tempo real os dados adquiridos pelo sistema.

Figura 33: Tela de acompanhamento das aquisições de cada dispositivo

Dispositivo 1 - Sensor de Temperatura Corporal

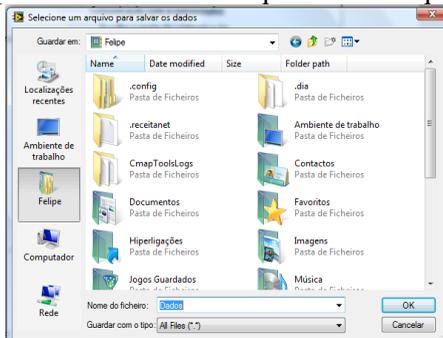


Fonte: Produção do próprio autor.

Para que seja possível armazenar os dados adquiridos para uma posterior avaliação, o usuário do sistema pode também utilizar o botão “Salvar Dados” disponível na tela inicial do *software*. Pressionado este botão o *software* irá armazenar os dados adquiridos por cada um dos dispositivos da rede, em um arquivo texto e em local definido pelo usuário.

Este sistema de aquisição inicia o *software* desabilitado e para que seja feita a gravação dos dados o usuário do sistema deve pressionar o botão “Salvar Dados”. Ao fazer isto o botão ficará verde e uma tela irá abrir pedindo ao usuário que salve os dados em um arquivo específico, conforme mostrado na Figura 34.

Figura 34: Tela para salvar os dados adquiridos em arquivo texto



Fonte: Produção do próprio autor.

O arquivo salvo terá um cabeçalho que informa de qual dispositivo são os dados e também cada leitura será acompanhada por uma aquisição de data e hora, para que seja possível se ter também um acompanhamento dos dados no tempo. O formato do arquivo salvo é mostrado na Figura 35. Este arquivo pode ser visualizado novamente através do *software* do Labview, ou mesmo através de *softwares* como Word ou Excel.

Figura 35: Formato do arquivo de dados salvo pelo *software*

Data Dispositivo 1	Dispositivo 2	Dispositivo 3
17/01/2013 10:26:09	0,435283	0,816872
17/01/2013 10:26:10	0,070320	0,918581
17/01/2013 10:26:11	0,607541	0,865664
17/01/2013 10:26:12	0,650745	0,622084
17/01/2013 10:26:12	0,476208	0,335683
17/01/2013 10:26:13	0,512578	0,629872
17/01/2013 10:26:14	0,625289	0,408070
17/01/2013 10:26:15	0,401166	0,806944
17/01/2013 10:26:15	0,510430	0,109465
17/01/2013 10:26:16	0,238446	0,403944
17/01/2013 10:26:17	0,499550	0,120975
17/01/2013 10:26:18	0,685987	0,603870
17/01/2013 10:26:18	0,890683	0,490812
17/01/2013 10:26:19	0,514998	0,923062
17/01/2013 10:26:20	0,250671	0,420713
17/01/2013 10:26:21	0,724700	0,694201
17/01/2013 10:26:21	0,667453	0,725902
17/01/2013 10:26:22	0,283092	0,751966
17/01/2013 10:26:23	0,208275	0,221654
17/01/2013 10:26:24	0,920728	0,325112
17/01/2013 10:26:24	0,630395	0,588707
17/01/2013 10:26:25	0,622212	0,491542
17/01/2013 10:26:26	0,904446	0,689465
17/01/2013 10:26:27	0,497987	0,064244
17/01/2013 10:26:27	0,727526	0,538781
17/01/2013 10:26:28	0,849022	0,700708

Fonte: Produção do próprio autor.

3.3 TESTES DO SOFTWARE DE INTERFACE COM O USUÁRIO

3.3.1 Teste de identificação dos dispositivos da rede

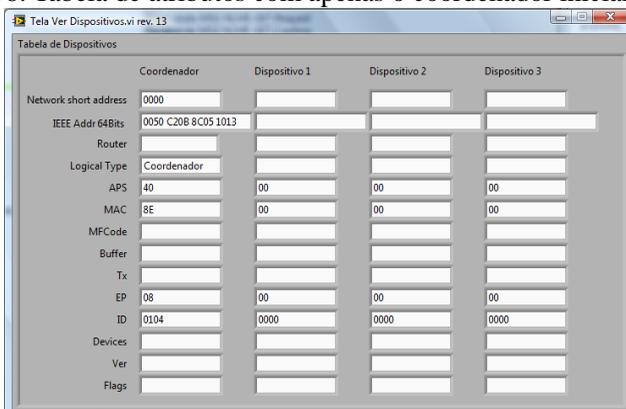
O objetivo deste teste é avaliar o *software* de interface com usuário, quanto à identificação de novos dispositivos presentes na rede.

A metodologia utilizada para esta avaliação foi inicializar o programa de interface com o usuário apenas com o coordenador conectado e verificar a tabela de atributos dos dispositivos mostrada por este mesmo *software*. Após isso se inicializar outros dois dispositivos, os

conectar a rede e verificar novamente a tabela de atributos. Para simplificar o teste, apenas alguns dos atributos serão verificados, como o endereço de rede (*Network short address*), a identificação do dispositivo (*IEEE Addr 64Bits*), o tipo lógico do dispositivo (*Logical Type*), e as identificações de APS, MAC, *endpoint* (EP) e *device ID* (ID).

Os resultados da tabela de atributos com o coordenador inicializado é mostrada na Figura 36. Nela verificou-se realmente que apenas os atributos de endereço de rede (*Network short address*), identificação do dispositivo (*IEEE Addr 64Bits*), tipo lógico do dispositivo (*Logical Type*), e as identificações de APS, MAC, EP e ID do coordenador estão preenchidos, conforme esperado.

Figura 36: Tabela de atributos com apenas o coordenador inicializado



	Coordenador	Dispositivo 1	Dispositivo 2	Dispositivo 3
Network short address	0000			
IEEE Addr 64Bits	0050 C20B 8C05 1013			
Router				
Logical Type	Coordenador			
APS	40	00	00	00
MAC	8E	00	00	00
MFCcode				
Buffer				
Tx				
EP	08	00	00	00
ID	0104	0000	0000	0000
Devices				
Ver				
Flags				

Fonte: Produção do próprio autor.

O resultado da tabela de atributos com o coordenador e mais dois dispositivos inicializados é mostrada na Figura 37. Nela novamente, conforme esperado verificamos que os atributos estão preenchidos para os três dispositivos presentes, deixando o dispositivo 3, que seria um quarto dispositivo possível na rede sem nenhum valor atribuído.

Figura 37: Tabela de atributos com o coordenador e mais dois dispositivos

	Coordenador	Dispositivo 1	Dispositivo 2	Dispositivo 3
Network short address	0000	796F	7970	
IEEE Addr 64Bits	0050 C208 8C05 1013	0050 C211 DC05 1801	0050 C203 5408 8C0F	
Router				
Logical Type	Coordenador	End Device	End Device	
APS	40	40	40	00
MAC	8E	8C	8C	00
MFCode				
Buffer				
Tx				
EP	08	08	08	00
ID	0104	0104	0104	0000
Devices				
Ver				
Flags				

Fonte: Produção do próprio autor.

Após a avaliação realizada pode ser concluído que o *software* está executando as rotinas de identificação dos dispositivos conectados na rede conforme projetado. O *software* está ainda realizando a verificação de quais atributos estão disponíveis em cada dispositivo e os armazenando na memória para que seja possível um correto gerenciamento do sistema de aquisição de dados.

3.3.2 Teste de aquisição de dados dos dispositivos

O objetivo deste teste é avaliar o *software* de interface com usuário, qual a capacidade de aquisição dos dados dos dispositivos presentes na rede sem fio *Zigbee*.

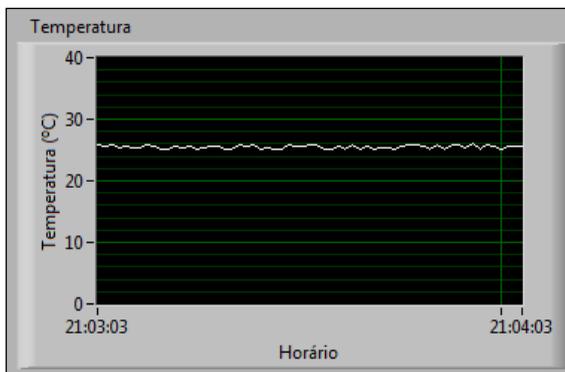
A metodologia para a execução deste teste foi inicializar a rede normalmente e adquirir o sinal do sensor de temperatura presente no *kit* de desenvolvimento. Para isto o dispositivo sensor foi configurado de forma a transmitir este sinal através do canal AD6. Este sensor é o LM61, e é alimentado com uma tensão de 3,3 V, gerando um sinal de saída na faixa de 0,3 V a 1,6 V. Possui uma sensibilidade de 10mV/°C, podendo atuar na faixa de -30 °C a 100 °C, com um erro aproximado de ± 3 °C para a faixa de temperaturas de -25 °C a 85 °C.

Este sinal foi lido pelo dispositivo sensor e enviado para o coordenador da rede e a na sequência para a interface com o usuário, onde foi monitorado e mostrado em um gráfico.

Os resultados desta avaliação são mostrados nos gráficos da Figura 38 e na Figura 39, mostram o acompanhamento do sinal de temperatura medida através do sensor presente no *kit* de desenvolvimento. Em ambos os gráficos, tem-se 60 pontos de temperatura, os quais foram adquiridos a uma taxa de 1 amostra por segundo.

Na Figura 38, é apresentado o gráfico com a medição da temperatura ambiente. O valor ficou estável e em torno de 25 °C, conferindo com a medição realizada em outro termômetro disponível no ambiente.

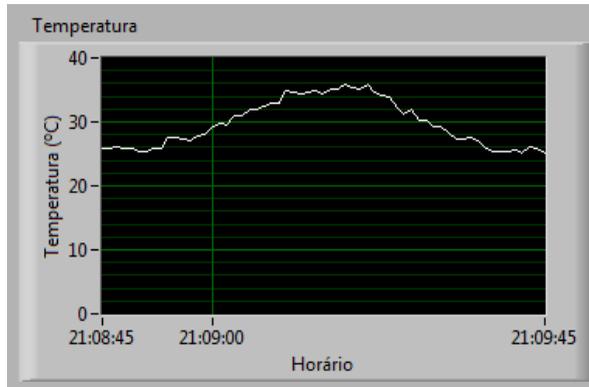
Figura 38: Dados adquiridos no teste de monitoramento do sensor de temperatura



Fonte: Produção do próprio autor.

Na Figura 39, a medição foi realizada também em temperatura ambiente, contudo, para se provocar um distúrbio na medição, foi colocado o dedo sobre o sensor de temperatura no *kit* de desenvolvimento. Com isso o valor esperado da medição deve ser algo próximo à temperatura do corpo humano, que é aproximadamente 36 °C. Nesta mesma figura pode ser percebido a perturbação no sinal medido, com o valor passando de aproximadamente 26 °C para um valor próximo a 35 °C. Após retirado o distúrbio do sensor, é possível notar que o valor medido da temperatura retornou para aproximadamente 26°C.

Figura 39: Monitoramento da temperatura quando aplicado um distúrbio no sensor.



Fonte: Produção do próprio autor.

Com as avaliações realizadas foi possível observar que o sistema realizou o monitoramento do sinal de temperatura da placa, conforme esperado por projeto, apresentando constância na medição do sinal quando este se manteve constante, e sendo capaz de perceber e registrar as variações do sinal quando um distúrbio foi aplicado.

3.3.3 Teste de geração de relatório de dados

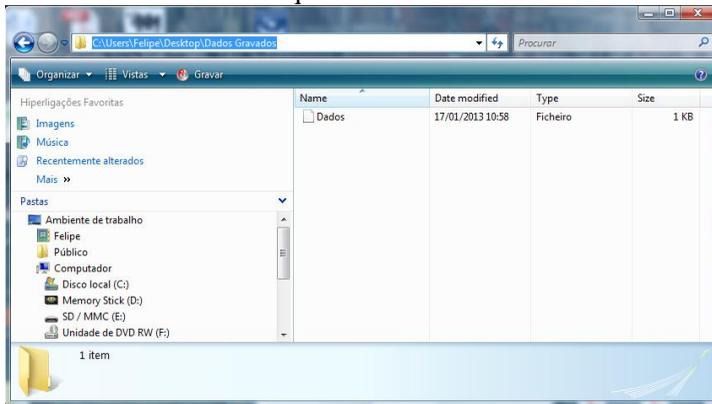
O objetivo desta avaliação é testar o *software* de interface com usuário, quanto capacidade de geração de relatório contendo os dados adquiridos pelos sistemas de aquisição para seu armazenamento e posterior tratamento.

A metodologia aplicada para a realização do teste consiste em inicializar o *software* com a opção de gravação de dados habilitada e verificar se o arquivo foi salvo com o nome correto, no local correto e com o formato correto, conforme explicado durante a descrição desta função dentro do capítulo 3.

Nesta avaliação é especificado que os dados sejam gravados na pasta “Dados Gravados”, salva na área de trabalho do computador, e que o arquivo deve ter o nome “Dados”.

Os resultados obtidos nesta avaliação são mostrados na Figura 40. Esta figura mostra a visualização do Windows Explorer, indicando que a pasta foi criada no local correto e que o arquivo com os dados foi salvo dentro desta pasta, conforme esperado.

Figura 40: Pasta e nome do arquivo salvo



Fonte: Produção do próprio autor.

A Figura 41 mostra o arquivo Dados.txt, salvo neste teste. Verifica-se que a formatação dos sinais adquiridos está correta dentro do arquivo.

Figura 41: Arquivo gerado no teste de gravação de dados

Dados sistema de aquisição			
Data	Dispositivo 1	Dispositivo 2	Dispositivo 3
17/01/2013 10:58:24	0,605421	0,043357	0,951503
17/01/2013 10:58:24	0,707676	0,121458	0,890963
17/01/2013 10:58:25	0,014076	0,094545	0,114099
17/01/2013 10:58:26	0,242766	0,039496	0,132651
17/01/2013 10:58:27	0,129326	0,519272	0,128590
17/01/2013 10:58:27	0,942107	0,328617	0,544927
17/01/2013 10:58:28	0,265296	0,218851	0,619208
17/01/2013 10:58:29	0,880368	0,837907	0,201498
17/01/2013 10:58:30	0,618611	0,828937	0,305724
17/01/2013 10:58:30	0,212478	0,136173	0,424171
17/01/2013 10:58:31	0,839106	0,701051	0,675093
17/01/2013 10:58:32	0,024171	0,533136	0,915591
17/01/2013 10:58:33	0,070277	0,219268	0,712038
17/01/2013 10:58:33	0,803026	0,364894	0,318843

Fonte: Produção do próprio autor.

Ao final desta avaliação pode ser concluído que o *software* está realizando a função de gravação de dados da maneira correta, permitindo ao usuário o armazenamento dos dados em arquivo texto conforme o proposto no início do trabalho.

CAPITULO 4 – SINAIS BIOELÉTRICOS

A automação de sistemas vem se disseminando muito nos últimos anos, estando presente em nosso dia a dia quando fazemos compras, vamos ao banco ou mesmo estamos em nossas casas ou trabalho. Neste contexto é crescente o número de oportunidades de se inserir novas tecnologias e automação também na área da medicina, onde o monitoramento de pacientes torna diagnósticos mais fáceis e precisos, melhorando ainda, a ergonomia, principalmente quando utilizados com redes de transmissão de dados sem fios.

Existem diversos sinais monitoráveis no corpo humano, que diferem entre si em termos de frequência, amplitude e forma de aquisição. A Tabela 14 mostra alguns sinais, comumente monitorados, com suas respectivas ordens de grandeza da medição e frequência, além da forma de sensoriamento utilizada.

Observando a forma de sensoriamento, nota-se que existem vários dispositivos transdutores disponíveis no mercado, resultantes de pesquisas ao longo dos últimos anos e que com a popularização da eletrônica e de redes sem fio podem ser melhor aproveitados no dia a dia das pessoas com algum tipo de enfermidade, principalmente nos procedimentos não invasivos, ou seja, que podem ser monitorados externamente ao corpo humano.

Medidas como respiração, batimentos cardíacos, oxigenação e pressão são imprescindíveis no monitoramento de pacientes que se encontram em quadros graves de saúde. Neste sentido é fundamental que no caso de qualquer tipo de alteração nessas medidas para níveis não aceitáveis seja tomada uma ação rápida pela equipe médica. Estes sinais possuem características de medição não invasivas, o que facilita sua aquisição mesmo sem a presença de pessoa especializada.

Nota-se também que grande parte destes sinais apresenta característica de sinais “fracos”, na ordem de milivolts, os quais precisam ser amplificados de forma que um sinal elétrico ou de origem biológica tenha uma amplitude que possa ser posteriormente processada, gravada e mostrada por dispositivos eletrônicos. Por questões de facilidade proporcionada pelas tecnologias disponíveis normalmente esses amplificadores são amplificadores de tensão.

Tabela 14: Diferentes sinais bioelétricos.

Parâmetro ou técnica de medição	Faixa da Medida Principal	Faixa da frequência do sinal (Hz)	Método de sensoriamento padrão
Balistocardiografia (BCG)	0-7mg	dc-40	Acelerômetro, Extensometro
	0-100 μ m	dc-40	LVDT - transformador diferencial de linearidade variável
Pressão da bexiga	1-100 cm H ₂ O	dc-10	Extensometro
Fluxo Sanguíneos	1-300 ml/s	dc-20	Medidor de fluxo (eletromagnético ou ultrasonico)
Pressão sanguínea / arterial (direta)	10-400mmHg	dc-50	Extensometro
Pressão sanguínea / arterial (indireta)	25-400mmHg	dc-60	Esfigmomanômetro
Pressão sanguínea / venosa	0-50mm Hg	dc-50	Extensometro
Gases Sanguíneos PO ₂	30-100mm Hg	dc-2	Eletrodo específico, volumétrico ou manométrico
Gases Sanguíneos PCO ₂	40-100mm Hg	dc-2	Eletrodo específico, volumétrico ou manométrico
Gases Sanguíneos PN ₂	1-3mm Hg	dc-2	Eletrodo específico, volumétrico ou manométrico
Gases Sanguíneos PCO	0,1-0,4mm Hg	dc-2	Eletrodo específico, volumétrico ou manométrico
pH Sanguíneo	6,8-7,8 unidade de pH	dc-2	Eletrodo específico
Débito cardíaco	4-25 litros/min	dc-20	Diluição de corante
Eletrocardiografia (ECG)	0,5-4mV	0,01-250	Eletrodos de pele
Eletroencefalografia (EEG)	5-300 μ V	dc-150	Eletrodo de cabeça
Eletrogastrografia (EGG)	10-1000 μ V	dc-1	Eletrodos de pele
	0,5-80mV	dc-1	Eletrodo de superfície estomacal
Eletromiografia (EMG)	0,1-5mV	0,1 - 1000	Eletrodos de superfície
Eletrooculograma (EOG)	50-3500 μ V	dc-50	Eletrodos de contato
Eletroretinograma (ERG)	0-900 μ V	dc-50	Eletrodos de contato
Resistência Galvânica da pele (GSR)	1-500k Ω	0,01-1	Eletrodos de pele
pH Gástrico	3-13 unidade de pH	dc-1	Eletrodo de pH
Pressão gastrointestinal	0-100 cm H ₂ O	dc-10	Estensometro
Força gastrointestinal	1-50g	dc-1	Medição de deslocamento, LVDT
Potencial de ação nervosa	0,01-3mV	dc-10000	Eletrodos de superfície ou agulhas
Fonocardiografia	Range dinâmico 80dB, limite 100 μ Pa	'5-2000	Microfone
Fluxo de Ar	0-600 litros/min	dc-40	Pressão diferencial
Taxa respiratória	2-50 respirações/min	0,1-10	Estensometro no peito, impedância e termistor nasal
Temperatura Corporal	32-40°C	dc-0,1	Termistor, termopar

Fonte: Adaptado de Webster, 2010.

Como característica básica os amplificadores de biopotenciais devem apresentar alta impedância de entrada para que não influenciem no sinal a ser medido. Os amplificadores também devem apresentar isolamento e circuitos de proteção que impeçam que correntes ou potenciais não desejados provoquem choques elétricos no paciente. No projeto de um amplificador deve-se limitar a banda de operação a um valor de frequência próximo a frequência do sinal a ser medido, de forma a possibilitar o correto processamento do sinal e obtendo uma boa relação sinal ruído. Outro ponto importante é que os amplificadores devem ser projetados de forma a proporcionar um fácil ajuste de calibração.

No presente trabalho foram escolhidos para serem monitorados pelo sistema de aquisição de dados três sinais principais: oxigenação do sangue, monitoramento de batimentos cardíacos e monitoramento da temperatura corporal.

Estes sinais foram escolhidos por serem comumente utilizados para o monitoramento de pacientes, sendo que os sinais de batimentos cardíacos e temperatura corporal são considerados sinais vitais ao corpo humano. Estes três sinais serão detalhados na sequência deste capítulo.

4.1 SISTEMA DE MONITORAMENTO DA OXIGENAÇÃO DO SANGUE

BAILEY (2008) relata que o oxigênio é um gás essencial para a vida humana. O transporte de oxigênio através do corpo humano é realizado pelo sistema circulatório, mais especificamente pelas hemoglobinas nas células vermelhas. Informações críticas podem ser obtidas através da medição da quantidade de oxigênio no sangue, como um percentual de sua capacidade máxima.

QURESHI (2009) comenta que para pacientes em risco de insuficiência respiratória, é importante monitorar a saturação de oxigênio do sangue para garantir a adequada perfusão de sangue no seu sistema. Preferencialmente a saturação de oxigênio deve ser recebida de forma contínua e monitorada através de métodos não invasivos de oximetria de pulso. Atualmente este tipo de monitoramento já se faz presente hospitais e clínicas, no entanto por meio de tecnologias com fios, as quais, vinculam este tipo de monitoramento a um indivíduo e a uma área específica, limitando a flexibilidade do sistema.

BAILEY (2008) explica ainda, que o oxímetro de pulso virou um procedimento padrão para a medição da saturação da oxigenação do

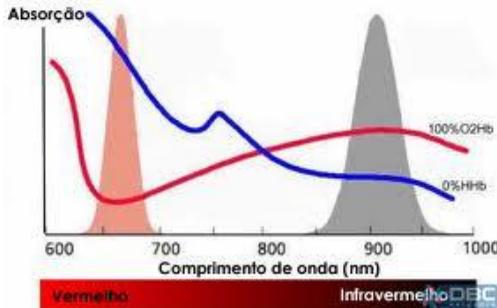
sangue nas salas de operação e recuperação hospitalares. A oximetria diminui o tempo para a detecção de hipoxemia, que é a baixa concentração de oxigênio no sangue arterial.

CARRARA (2009) aduz que o oxímetro de pulso arterial permite medir continuamente e de maneira não invasiva a saturação o oxigênio (SpO_2) da hemoglobina arteriolar, em uma região anatômica que permita a aferição da medida, preferencialmente de localização periférica, como extremidades digitais das mãos e dos pés, mãos, pés, lóbulo da orelha, dentre outros. É utilizada para monitorar pacientes com risco de desenvolver hipoxemia, pois a observação do paciente, mesmo realizada por profissionais experientes, não é capaz de precisar tal enfermidade. Até que a SpO_2 arterial esteja abaixo de 80%. CARRARA (2009), comenta que a monitoração da SpO_2 fornece informação acerca dos sistemas cardíaco, respiratório e do transporte de oxigênio no organismo. É amplamente utilizado por ser não invasiva, monitorando de maneira contínua, além de ser de instalação simples e indolor.

CARRARA (2009) afirma que o princípio da oximetria de pulso arterial é baseado na hemoglobina oxigenada (saturada) e reduzida. Este princípio é denominado espectrofotometria. Isto se dá porque o sangue saturado de oxigênio tem um espectro de absorção de luz diferente do sangue não saturado de oxigênio. Assim, a quantidade de luz no espectro vermelho e infravermelho absorvida pelo sangue, pode ser utilizada para calcular a taxa da hemoglobina oxigenada em relação a hemoglobina total no sangue arterial, sendo apresentada no monitor do equipamento como a percentagem de SpO_2 . Os valores normais situam-se, normalmente, entre 95 a 100%.

A oxihemoglobina ou hemoglobina saturada absorve mais luz infravermelha e permite que a luz vermelha passe por meio dela. A luz infravermelha se encontra na faixa de 850 a 1000 nanômetros (nm). A desoxihemoglobina, ou hemoglobina reduzida absorve mais a luz vermelha e permite que mais luz infravermelha passe através dela. A luz vermelha se encontra na faixa de comprimento de onda de 600 a 750 nanômetros (nm). A Figura 42 mostra a absorção da hemoglobina em cada comprimento de onda.

Figura 42: Absorção da hemoglobina VS comprimento de onda.

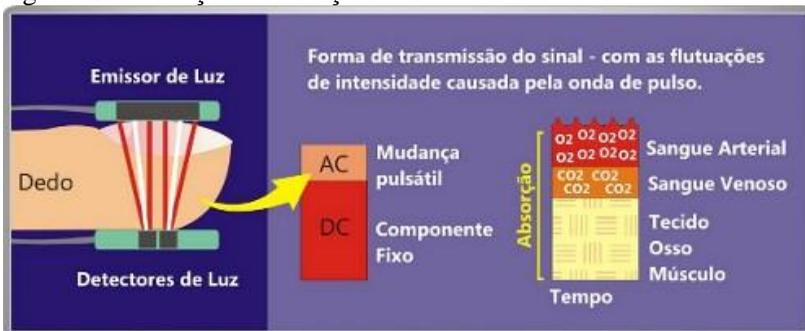


Fonte: Adaptado de DBC (2013).

Com isso, utilizando diodos emissores de luz e fotodetectores em dois comprimentos de onda distintos (um no infravermelho, outro em vermelho) e circuitos eletrônicos (conversor de corrente/tensão, filtros e amplificadores) é possível obter um sinal pulsante que pode ser processado para obter uma leitura de saturação de oxigênio.

A Figura 43 mostra como é feita a medição da oxigenação do sangue através do dedo do paciente. Basicamente a luz vermelha e infravermelha é emitida e detectada novamente, esta diferença é medida e com isso se determina o percentual de oxigênio presente na corrente sanguínea. A atenuação do sinal medido tem uma componente fixa, a qual é atribuída aos tecidos, ossos ou músculos, além do sangue venoso e uma componente pulsante, atribuída a diferença de atenuação no sangue arterial.

Figura 43: Ilustração da medição da oximetria no dedo.



Fonte: Adaptado de Tetzlaff AAS (2010).

O oxímetro de pulso analisa a absorção da luz dos dois comprimentos de onda na parte pulsada do sangue arterial e calcula a razão da absorção usando a Eq. 1.

$$SpO_2 = \frac{\frac{AC_{660}}{DC_{660}}}{\frac{AC_{940}}{DC_{940}}} \quad \text{Eq. 1}$$

Onde:

AC_{660} = Parte alternada da absorção no comprimento de onda de 660 nm

DC_{660} = Parte contínua da absorção no comprimento de onda de 660 nm

AC_{940} = Parte alternada da absorção no comprimento de onda de 940 nm

DC_{940} = Parte contínua da absorção no comprimento de onda de 940 nm

LOPEZ (2012) comenta que o valor de SpO_2 é retirado de uma tabela calculada através de fórmulas empíricas. Uma razão de 1 representa um valor de SpO_2 de 85%, uma razão de 0,4 representa SpO_2 de 100% e uma razão de 3,4 representa um valor de SpO_2 de 0%. Para maior precisão a tabela deve ser baseada em medições de experimentos de pacientes saudáveis.

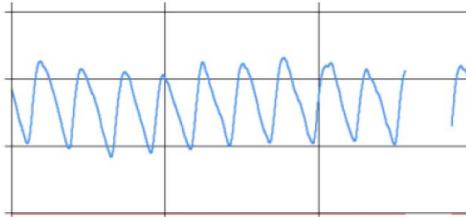
Outra forma para calcular o valor de SpO_2 é retirando o valor apenas da componente pulsante e utilizando a Eq. 2.

$$R = \frac{\log_{10}(I_{AC})\lambda_1}{\log_{10}(I_{AC})\lambda_2} \quad \text{Eq. 2}$$

Onde I_{ac} é a intensidade da luz em 1 (660nm) ou 2 (940nm), onde apenas o valor pulsado está presente.

A Figura 44 mostra um sinal típico de um oxímetro de pulso. Este sinal representa a absorção do sangue arterial pulsado. Pode-se ainda, através deste sinal, calcular o número de batimentos cardíacos por minuto.

Figura 44: Sinal típico de um oxímetro de pulso



Fonte: LOPEZ (2012)

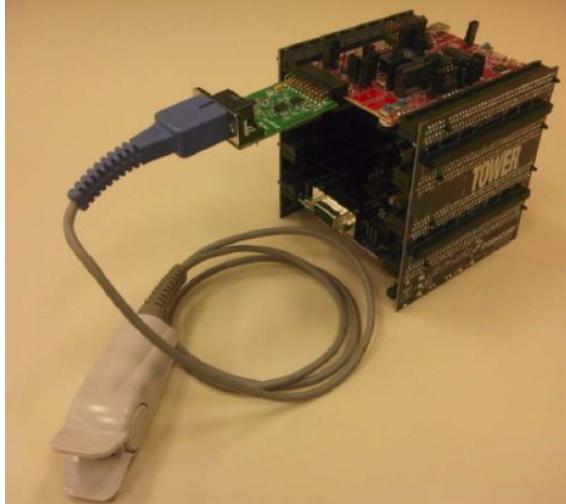
4.1.1 Circuito utilizado para a aquisição de dados de oxigenação do sangue

O objetivo deste trabalho é a transmissão de sinais bioelétricos através de uma rede sem fios *Zigbee*. Com isso não se buscou desenvolver um circuito específico para os sensores biomédicos, mas sim utilizar sensores comerciais.

Para o sensor de oximetria foi utilizado além do *kit Zigbee* já abordado neste trabalho, o *kit* de desenvolvimento da *Freescale* “*MED-SPO2: Pulse Oximeter Development Kit*” para fazer a aquisição dos dados de oxigenação do sangue. Este *kit* foi adaptado de forma a enviar os valores lidos de oxigenação do sangue para o *kit Zigbee* através de comunicação serial. Por fim o *kit Zigbee* faz o envio dos dados de oxigenação para o dispositivo coordenador da rede através do protocolo *Zigbee* e o coordenador da rede disponibiliza o dado para a interface com o usuário.

O *kit* de desenvolvimento *MED-SPO2* é mostrado na Figura 45. Este *kit* utiliza a base *Tower* da *Freescale* e é implementado utilizando o microcontrolador *Kinetis K53*, embutido com funcionalidades importantes para o tratamento de sinais para oximetria e outras medições de sinais em aplicações médicas.

Figura 45: *Kit de desenvolvimento MED-SPO2 da Freescale*



Fonte: Produção do próprio autor.

Dentre as funções disponibilizadas pelo Kinetis K53, destacam-se:

- 32-bit ARM® Cortex™-M4
- Instruções DSP para filtro de sinais
- Dois Amplificadores Operacionais
- Dois Amplificadores de Transimpedância

O *kit* Med-SPO2 utiliza ainda o sensor comercial Nelcor DS100-A, do fabricante Corvidien. Este sensor é comumente utilizado para medição de oximetria em diversos equipamentos hospitalares comerciais.

A comunicação entre o *kit* Zigbee e o *kit* do oxímetro será feita utilizando um protocolo serial. A Figura 46 mostra o formato do protocolo de dados utilizado para esta comunicação.

Figura 46: Formato do pacote de comunicação serial.

Tipo do Pacote	Opcode do Comando	Tamanho do Dado	Dado 0	Dado n
----------------	-------------------	-----------------	--------	------	--------

Fonte: Adaptado de LOPEZ, 2012

O primeiro *byte* chamado de “Tipo do Pacote” é o que define o tipo do pacote de dados recebido. Para a aplicação do oxímetro de pulso apenas três tipos de pacotes estão disponíveis, tais como:

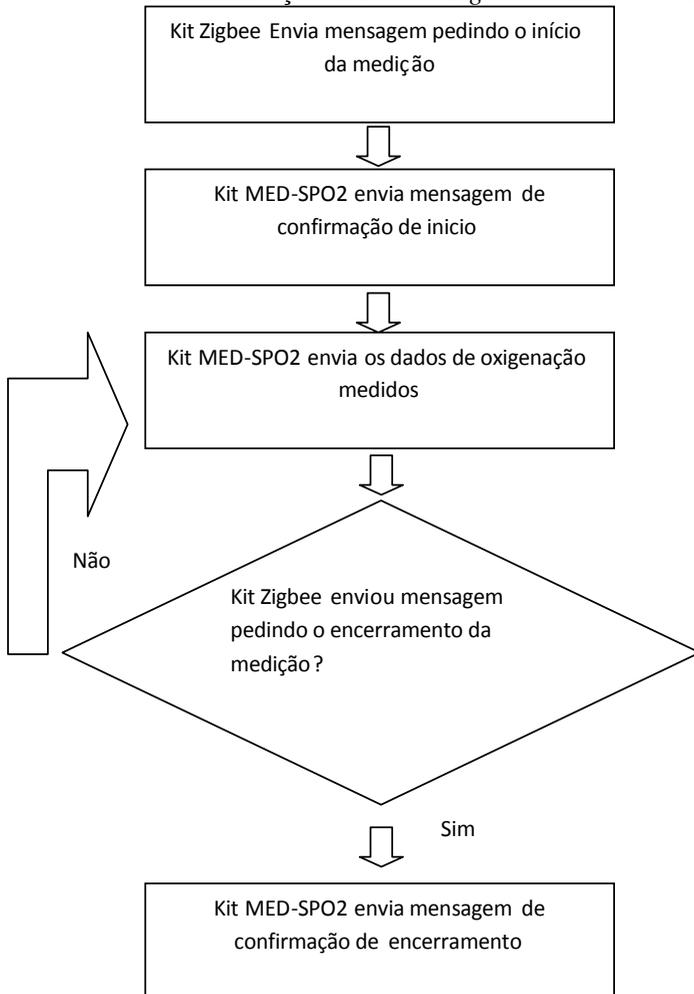
- *Request* (REQ) – Faz a requisição de um dado
- *Confirmation* (CFM) – Faz a confirmação do correto recebimento das mensagens
- *Indication* (IND) – Faz a indicação dos dados requisitados.

Se o primeiro *byte* é reconhecido como um comando *Request* (REQ), o segundo *byte*, chamado de “Opcode do Comando”, é verificado para determinar qual o tipo de comando deve ser respondido. A aplicação do oxímetro responde a quatro comandos:

- *SpO2StartMeaserementReq*
- *SpO2AbortMeaserementReq*
- *SpO2DiagnosticModeStartMeaserementReq*
- *SpO2DiagnosticModeStopMeaserementReq*

A Figura 47 mostra o fluxo da comunicação entre o *kit Zigbee* e o *kit* do oxímetro. Basicamente, o *kit Zigbee* pede o início da medição do oxímetro. Na sequencia o *kit* do oxímetro envia uma mensagem confirmando o início da medição e em seguida inicia o envio dos dados de oxigenação. Isto acontece até o *kit Zigbee* envie uma mensagem pedindo pelo encerramento da medição.

Figura 47: Fluxo da comunicação entre o *kit Zigbee* e o *Kit* do Oxímetro



Fonte: Produção do próprio autor.

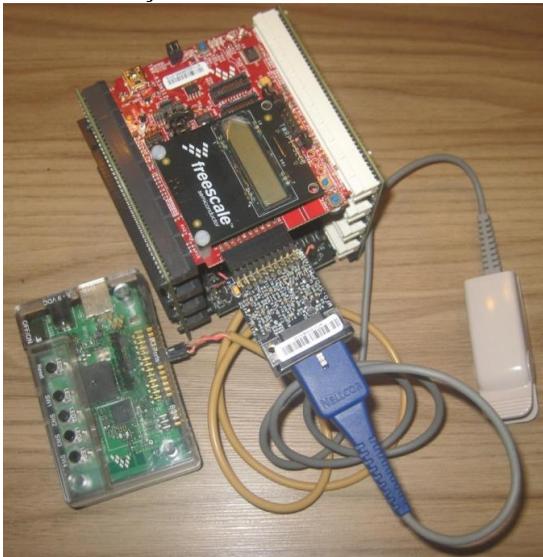
O protocolo de comunicação completo disponibilizado para o *kit* de desenvolvimento MED-SPO2 está disponível em LOPEZ (2012). Neste documento foi possível verificar todas as mensagens e parâmetros necessários para a implementação realizada.

Devido ao fato da medição de oxigenação no sangue não ser uma grandeza que varia significativamente com o tempo e de acordo com *ZIGBEE HEALTH CARE PROFILE SPECIFICATION* (2010),

normalmente tem uma frequência de medição típica de algumas vezes ao dia, o *kit Zigbee* foi programado de forma a pedir o valor de oxigenação para o *kit* do Oxímetro a cada 30 segundos. Para isso foi programado um temporizador dentro do *software* do *kit Zigbee* que gera uma interrupção, na qual é enviada a mensagem pedindo o valor da oxigenação no sangue para o *kit* do oxímetro.

A Figura 48 mostra o *kit Zigbee* utilizado neste trabalho acoplado com o *kit* de desenvolvimento utilizado para a aquisição do sinal de oxigenação do sangue. A conexão e comunicação entre os dois *kits* de desenvolvimento é feita através de comunicação serial entre os dois microcontroladores.

Figura 48: Foto do *kit* de medição de oxigenação no sangue acoplado ao *kit Zigbee*, via comunicação serial.



Fonte: Produção do próprio autor.

4.1.2 Teste de medição da oxigenação do sangue

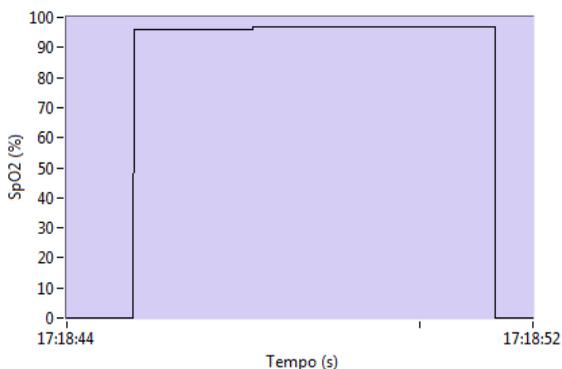
O objetivo deste teste é avaliar o sistema de medição de oxigenação do sangue, desde o circuito eletrônico para a aquisição dos dados, até o *software* que faz o monitoramento e mostra os dados para o usuário do sistema.

Para isto a metodologia do teste consiste em iniciar a rede

Zigbee com o dispositivo sensor de oxigenação do sangue, posicionar o dispositivo sensor de oxigenação no sangue no dedo indicador do voluntario e iniciar a aquisição de dados. O sistema de aquisição está posicionado no mesmo ambiente do voluntario. Os dados serão então monitorados e dispostos nos gráficos da interface com o usuário, além de salvos em arquivo texto.

A Figura 49 mostra o gráfico da medição realizada no teste do sensor de oxigenação do sangue. Já a Figura 50 mostra o arquivo de dados gerado neste mesmo teste.

Figura 49: Resultados mostrados pela interface gráfica no teste de medição do sensor de oximetria



Fonte: Produção do próprio autor

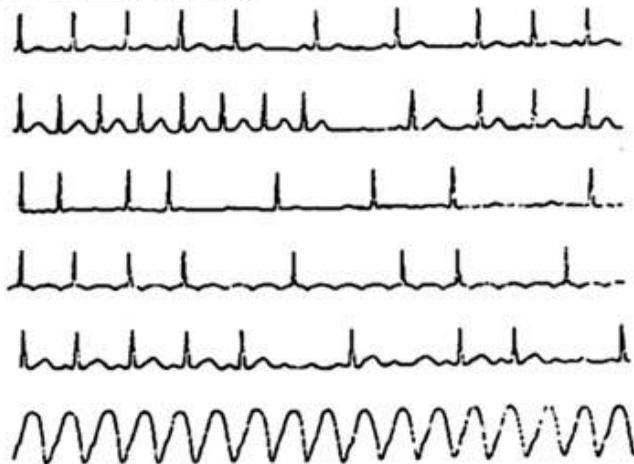
a parte periférica do aparelho circulatório e o próprio funcionamento do coração podem ser avaliados pela análise das pulsações arteriais, venosas e capilares.

PORTO (2008) afirma que a tomada do pulso radial é um ato simples, porem rico de significação. Além de propiciar ao médico informações semióticas, simboliza a relação médico paciente constituindo com frequência o primeiro contato direto entre eles. Ele explica ainda que é necessário contar sempre o número de pulsações durante um minuto inteiro, comparando-se estes valores com o número de batimentos cardíacos.

Segundo PORTO (2008), em pessoas adultas, considera-se normal a frequência de 60 a 100 batimentos por minuto, em repouso. Acima de 100 pulsações designa-se taquisfigmia ou, como é mais usado na linguagem comum, taquicardia. Em várias condições fisiológicas, como exercícios, emoção, gravidez, ocorre taquicardia. Quando ocorrem menos de 60 pulsações por minuto, dá-se o nome de bradisfigmia, ou bradicardia. Esta condição pode ocorrer em estados febris, hipertireoidismo, ou insuficiência cardíaca, entre outros.

É importante ainda a verificação do ritmo cardíaco. PORTO (2008) afirma que a irregularidade do pulso indica alteração do ritmo cardíaco – arritmia que pode ser fisiológica ou patológica. A Figura 51 mostra exemplos de diferentes ritmos do pulso.

Figura 51: Arritmias cardíacas.



Fonte: adaptado de CHAVEZ, 2013.

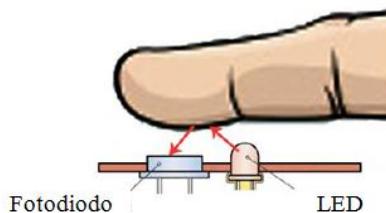
WOODWARD (2012), relata que a frequência cardíaca pode ser mensurada, de forma manual, em qualquer lugar do corpo onde pode ser detectada a pulsação arterial. A medição de pulso é muito utilizada durante a prática de exercícios, servindo como um parâmetro para medir a intensidade da atividade que está sendo executada.

4.2.1 Circuito utilizado para aquisição de dados de batimentos cardíacos

WOODWARD (2012) comenta que existem diversos métodos para detectar eletronicamente o batimento cardíaco de seres humanos. Os batimentos podem ser detectados acusticamente (estetoscópio ou doppler), mecanicamente (esfigmomanômetro), eletricamente (Eletrocardiograma) e opticamente. Este último se dá através do fato de que pequenos vasos sanguíneos subcutâneos em vários lugares da pele (ponta dos dedos, orelha, etc) alternadamente se expandem e se contraem sincronizadamente com o as batidas do coração.

O método utilizado neste trabalho se baseia na detecção de batimentos opticamente. Este sensor pode ser colocado no dedo ou no lobo da orelha e para a construção deste sensor são utilizados um Led e um fotodiodo, responsáveis pela geração de um sinal luminoso e pela detecção da quantidade de luz, respectivamente. A Figura 52 mostra o princípio de funcionamento quando o sensor está colocado no dedo.

Figura 52: Princípio de funcionamento do sensor de batimentos cardíacos



Fonte: Adaptado de SUNROM TECHNOLOGIES, 2013.

Basicamente este método se baseia na captura da diferença de luminosidade detectada pelo fotodiodo quando o sinal de luz atravessa, ou reflete no dedo ou lobo da orelha. A variação da luminosidade detectada pelo fotodiodo, se dá devido ao fato de que quando o coração

bombeia um pulso de sangue através dos vasos sanguíneos, o dedo se torna sensivelmente mais opaco e com isso menos luz atinge o detector. Esta variação pode então ser convertida em sinal elétrico e processada para se obter a taxa de batimentos cardíacos.

O sensor utilizado nesta implementação é mostrado na Figura 53. Este é um sensor comercial, modelo 114010, o qual pode ser posicionado no dedo ou lobo da orelha para se fazer a medição dos batimentos cardíacos. O sensor tem conexão USB e quando conectado ao computador mostra o nível dos batimentos cardíacos através de um *software* específico.

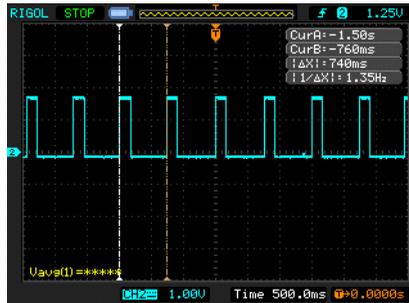
Figura 53: Sensor utilizado para o monitoramento dos batimentos cardíacos



Fonte: Produção do próprio autor.

Neste trabalho este sensor foi conectado ao *kit* de desenvolvimento *Zigbee*, o qual provê alimentação de 3 Vdc, e faz a aquisição do sinal de batimentos cardíacos. O sinal disponibilizado pelo sensor, apresentado na Figura 54, se dá na forma de pulsos, que variam em frequência de acordo com a taxa de batimentos cardíacos. Nota-se nesta figura que a frequência do sinal é 1,35 Hz, o que convertido para batimentos por minuto é igual a 81 bpm. O cálculo de frequência cardíaca é feito multiplicando-se o valor de frequência do sinal por 60.

Figura 54: Sinal medido no sensor de batimentos cardíacos utilizando um osciloscópio.



Fonte: Produção do próprio autor

Os pulsos são então adquiridos pela entrada AD do *kit* de desenvolvimento *Zigbee* e processado de forma a ser enviado para o dispositivo coordenador da rede e posteriormente para a interface com o usuário. Na interface com o usuário este sinal é então tratado de forma a se obter sua frequência e calcular então a taxa de batimentos por minuto, além de disponibilizar um gráfico do acompanhamento da taxa de batimentos cardíacos ao longo do tempo.

O circuito de medição de batimentos cardíacos adaptado para o uso através da rede sem fios *Zigbee* é mostrado na Figura 55.

Figura 55: Foto do circuito de medição de batimentos cardíacos acoplado ao *kit Zigbee*



Fonte: Produção do próprio autor

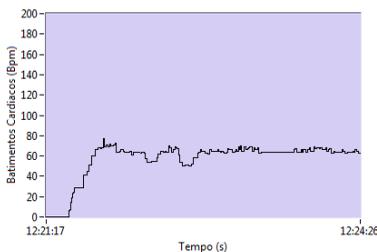
4.2.2 Teste de medição batimentos cardíacos

O objetivo deste teste é avaliar a medição do sinal do sensor batimentos cardíacos, analisando como este sinal é adquirido e disponibilizado para o usuário do sistema.

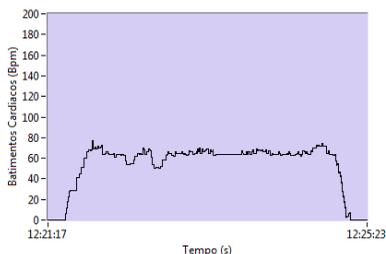
Para isto a metodologia utilizada consiste em iniciar normalmente com o dispositivo sensor de batimentos cardíacos. O sensor ficará inicialmente sem estar conectado ao voluntário. Neste momento o sinal lido deve ser igual a zero. Em seguida o sensor será posicionado no dedo indicador do voluntário e será realizada a medição dos batimentos cardíacos. O resultado esperado é o gráfico de monitoramento seguindo a taxa de batimentos lida pelo sensor corretamente e o arquivo de dados com todas as aquisições realizadas durante o teste. O sistema de aquisição estará posicionado no mesmo ambiente do voluntario, a poucos metros do mesmo.

A Figura 56 mostra dois gráficos da aquisição feita no teste do sistema de monitoramento de batimentos cardíacos. O gráfico (a), mostra os instantes iniciais do teste onde o valor do sinal é igual a zero. Neste momento o sensor ainda não estava posicionado no dedo do paciente e a leitura esperada era realmente zero. Ao e conectar o sensor no paciente o valor de batimentos começou a ser monitorado e após um período de estabilização alcançou um valor aproximado de 60 bpm. Este valor é coerente com o valor esperado para uma pessoa adulta em condição normal.

Figura 56: Resultados mostrados pela interface gráfica durante a aquisição de dados do teste de batimentos cardíacos.



a) Parte inicial do teste



b) Dados do teste completo

Fonte: Produção do próprio autor

Na Figura (b), que mostra os resultados completos do teste nota-se o momento onde o sensor foi novamente desconectado do paciente e o sistema voltou a medir o valor zero conforme esperado. A Figura 57 mostra a planilha texto com os dados que foram adquiridos e armazenados durante o teste de medição dos batimentos cardíacos.

Figura 57: Imagem do arquivo de dados gravado durante o teste de medição dos batimentos cardíacos.

Dados sistema de aquisição			
Data	Dispositivo 1	Dispositivo 2	Dispositivo 3
22/06/2013 12:23	64	0	0
22/06/2013 12:23	67	0	0
22/06/2013 12:23	64	0	0
22/06/2013 12:23	64	0	0
22/06/2013 12:23	66	0	0
22/06/2013 12:23	66	0	0
22/06/2013 12:23	66	0	0
22/06/2013 12:23	64	0	0
22/06/2013 12:23	66	0	0
22/06/2013 12:23	68	0	0
22/06/2013 12:23	65	0	0
22/06/2013 12:23	65	0	0
22/06/2013 12:23	67	0	0
22/06/2013 12:24	68	0	0
22/06/2013 12:24	68	0	0
22/06/2013 12:24	67	0	0
22/06/2013 12:24	68	0	0
22/06/2013 12:24	66	0	0
22/06/2013 12:24	66	0	0
22/06/2013 12:24	65	0	0
22/06/2013 12:24	65	0	0
22/06/2013 12:24	67	0	0
22/06/2013 12:24	63	0	0
22/06/2013 12:24	63	0	0
22/06/2013 12:24	64	0	0

Fonte: Produção do próprio autor

Os resultados obtidos na medição do sensor de batimentos cardíacos adaptado para o *kit* de comunicação *Zigbee* mostram que o sistema funcionou conforme o projetado, sendo capaz de fazer o monitoramento da taxa de batimentos cardíacos de forma consistente e gerando os gráficos e arquivos conforme especificado para o *software* de monitoramento.

4.3 SISTEMA DE MONITORAMENTO DA TEMPERATURA CORPORAL

ESTEVES (2010) descreve os sinais vitais como medidas de várias estatísticas fisiológicas procuradas para avaliar as funções corporais básicas, onde sua avaliação é fundamental para a determinação do bem-estar físico do indivíduo.

Consideram-se cinco sinais vitais:

- Tensão Arterial
- Frequência Cardíaca
- Frequência Respiratória
- Temperatura
- Dor

Segundo PORTO (2008), a temperatura do interior do corpo permanece quase constante, dentro de uma variação de no máximo 0,6°C, mesmo quando exposto a extremos de frio ou calor, graças ao aparelho termorregulador. A temperatura da parte externa do corpo, ao contrário, está sujeita às variações das condições ambientais. Além disso, pequenas variações na temperatura normal são observadas, de pessoa para pessoa, e numa mesma pessoa em diferentes locais do corpo.

PORTO (2008), comenta ainda que os valores da temperatura podem estar aumentados em certas condições, tais como exercícios intensos, quando a temperatura retal pode atingir 38 e até 40°C. Por outro lado, quando o corpo fica exposto a frio intenso, a temperatura retal pode cair para valores abaixo de 36°C.

O local para a verificação da temperatura corporal habitualmente utilizado é o oco axilar, contudo a temperatura pode ser também tomada na boca ou no reto. As temperaturas consideradas normais são mostradas na Tabela 15.

Tabela 15: Valores normais para a temperatura corporal.

Local	Valor (°C)
Temperatura Axilar	35,5 a 37
Temperatura Bucal	36 a 37,4
Temperatura Retal	36 a 37,5

Fonte: Adaptado de PORTO, 2008.

Quando a temperatura corporal atinge 37°C inicia-se a sudorese, que a partir deste ponto aumenta rapidamente com um mínimo de elevação na temperatura corporal. Quando o corpo é resfriado abaixo de 37°C, são postos em ação mecanismos especiais para conservar o calor corporal.

PIANCINI (2005) comenta que a hipertermia, ou febre, que é a temperatura corporal acima da faixa da normalidade pode causar transtornos ao cérebro e a outros órgãos do corpo humano. A febre pode

ser resultado de infecções, lesões teciduais, processos inflamatórios e neoplasias malignas, além de outras condições.

Os termômetros clínicos são normalmente de mercúrio graduado em quintos ou décimos, marcando de 35 a 42°C. Em geral a marca dos 37°C é de cor diferente, por se considerar este o valor máximo corporal.

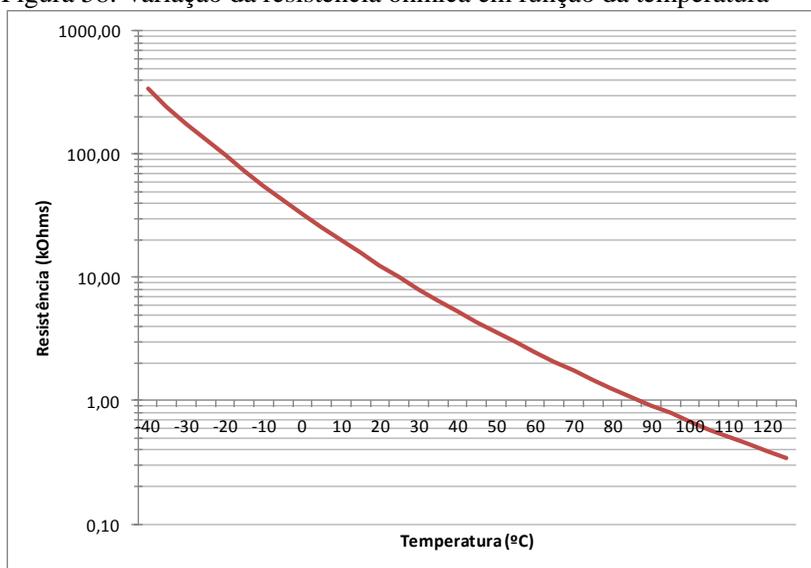
4.3.1 **Circuito utilizado para a aquisição da temperatura corporal**

O sistema de aquisição de dados para o sinal de temperatura corporal foi projetado para obter a máxima precisão na faixa de temperatura de 30°C a 45°C, visto que a temperatura de interesse na aplicação está na faixa de 35 a 42°C.

O *kit* de desenvolvimento possui um sensor de temperatura, contudo o mesmo não pode ser utilizado, pois é fixo na placa de circuito impresso. Com isso escolheu-se um sensor de temperatura externo para realizar a medição no paciente. Este sensor é do tipo NTC - *Negative Temperature Coefficient*, do fabricante EPCOS código K800/10k/1%. Este sensor apresenta um valor de resistência de 10k ohms à temperatura de 25°C.

A Figura 58 mostra a curva do sensor de temperatura. Para a faixa de -40°C a 125°C, nota-se que a resistência pode variar de valores de aproximadamente 350k ohms a 300 ohms.

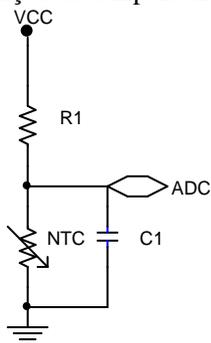
Figura 58: Variação da resistência ôhmica em função da temperatura



Fonte: Produção do próprio autor

O circuito de alimentação e aquisição do sinal do sensor é apresentado na Figura 59. Nesta figura, pode-se perceber que se utiliza a alimentação do *kit* de desenvolvimento (VCC), e que se tem um resistor (R1) para adequar o nível de tensão para a porta ADC do microcontrolador e que foi adicionado um filtro *anti-aliasing* (C1). O sinal de saída do sensor vai diretamente para a porta ADC do microcontrolador.

Figura 59: Circuito de medição da temperatura corporal



Fonte: Produção do próprio autor

A Tabela 16 mostra a especificação de resistência do NTC durante toda a sua variação de temperatura, assim como o valor de resistência máxima e mínima para cada uma destas temperaturas. A partir destes valores foi dimensionado o valor do resistor R1 de forma a deixar a leitura na faixa de 30 °C a 45 °C com a melhor resolução, pois esta é a faixa de interesse para esta aplicação.

Nesta tabela foi também calculado o valor da conversão analógica digital para as diferentes faixas de temperatura, considerando os 10bits de resolução do microcontrolador. Nota-se que para a faixa de 30 °C a 45 °C, que se tem com um resistor R1 de 4,7 k ohms aproximadamente 50 pontos a cada 5°C de temperatura, o que nos dá uma resolução de 0,1 °C, adequada a esta aplicação.

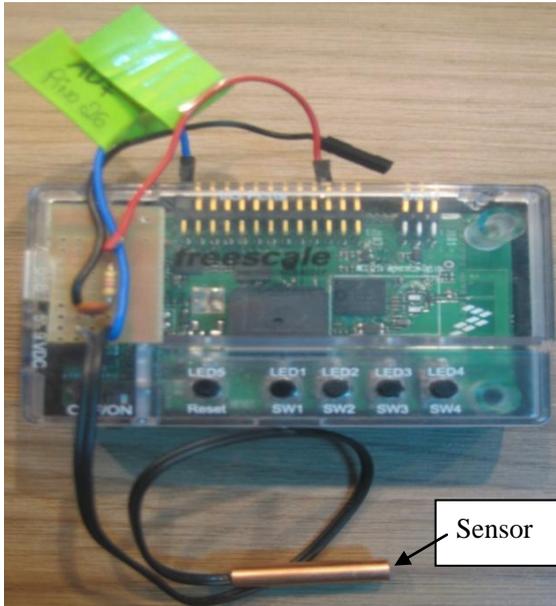
O valor para o capacitor C1 é de 100nF. A tensão VCC é 3 V e o valor esperado para o ADC na faixa de interesse de 30 °C a 45 °C é de 1,11 V a 1,55 V. O circuito mostrado na Figura 59 foi conectado na porta AD7 do microcontrolador e o conversor analógico digital foi configurado de forma a fazer uma leitura a cada cinco segundos. A Figura 60 mostra a foto do sistema implementado para fazer a medição da temperatura corporal.

Tabela 16: Curva do NTC e cálculo da precisão de leitura

Curva NTC							R1	VALOR ADC 10bits	Nº Pontos	
Temp [°C]	R Nom [Ω]	R Min [Ω]	R Max [Ω]	ΔR [±%]	ΔT [±°C]	α [%/K]	R Div [Ω]	Read Nom[pu]	Der [pu]	Resolução (°C)
-40	336500	320419	352582	4,8	0,7	6,7	4700	14	5	1,000
-35	242589	231867	253311	4,4	0,7	6,4	4700	19	7	0,714
-30	177000	169786	184214	4,1	0,7	6,2	4700	26	9	0,556
-25	130370	125487	135252	3,7	0,6	6	4700	35	12	0,417
-20	97070	93743	100397	3,4	0,6	5,8	4700	47	14	0,357
-15	72929	70652	75206	3,1	0,6	5,6	4700	61	19	0,263
-10	55330	53765	56895	2,8	0,5	5,4	4700	80	22	0,227
-5	42315	41237	43393	2,5	0,5	5,3	4700	102	26	0,192
0	32650	31907	33393	2,3	0,4	5,1	4700	128	31	0,161
5	25388	24877	25898	2	0,4	5	4700	159	36	0,139
10	19900	19550	20250	1,8	0,4	4,8	4700	195	40	0,125
15	15708	15470	15946	1,5	0,3	4,7	4700	235	44	0,114
20	12490	12330	12650	1,3	0,3	4,5	4700	279	48	0,104
25	10000	9900	10100	1	0,2	4,4	4700	327	50	0,100
30	8057	7955	8159	1,3	0,3	4,3	4700	377	51	0,098
35	6531	6434	6628	1,5	0,4	4,1	4700	428	51	0,098
40	5327	5237	5417	1,7	0,4	4	4700	479	51	0,098
45	4369	4286	4451	1,9	0,5	3,9	4700	530	49	0,102
50	3603	3528	3678	2,1	0,5	3,8	4700	579	47	0,106
55	2986	2918	3054	2,3	0,6	3,7	4700	626	43	0,116
60	2488	2427	2549	2,5	0,7	3,6	4700	669	40	0,125
65	2083	2028	2138	2,6	0,8	3,5	4700	709	36	0,139
70	1752	1703	1801	2,8	0,8	3,4	4700	745	33	0,152
75	1481	1437	1525	3	0,9	3,3	4700	778	29	0,172
80	1258	1219	1297	3,1	1	3,2	4700	807	26	0,192
85	1072	1037	1108	3,3	1	3,2	4700	833	23	0,217
90	917,7	886,1	949,3	3,4	1,1	3,1	4700	856	20	0,250
95	788,5	760,2	816,9	3,6	1,2	3	4700	876	18	0,278
100	680	654,6	705,4	3,7	1,3	2,9	4700	894	16	0,313
105	588,6	565,8	611,4	3,9	1,4	2,9	4700	910	13	0,385
110	511,2	490,7	531,7	4	1,4	2,8	4700	923	12	0,417
115	445,4	426,9	463,9	4,2	1,5	2,7	4700	935	10	0,500
120	389,3	372,6	406	4,3	1,6	2,7	4700	945	9	0,556
125	341,7	326,6	356,8	4,4	1,7	2,6	4700	954	0	-

Fonte: Produção do próprio autor

Figura 60: Foto do dispositivo com o sistema de aquisição de temperatura



Fonte: Produção do próprio autor

4.3.2 Teste de medição da temperatura corporal

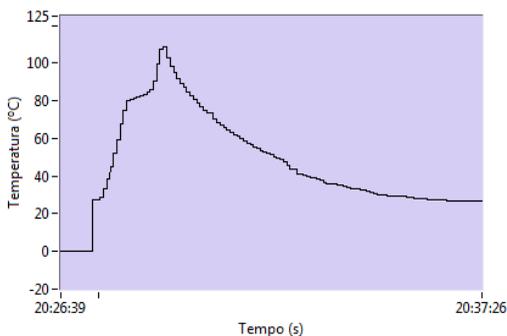
Este teste tem como objetivo avaliar a medição do sinal do sensor temperatura corporal, analisando como este sinal é adquirido e disponibilizado para o usuário do sistema.

Para este teste metodologia utilizada é de iniciar a rede *Zigbee* normalmente com o dispositivo sensor de temperatura. O teste irá partir da temperatura ambiente e elevar a temperatura lida pelo sensor até aproximadamente 100°C, retornando então para a temperatura ambiente novamente com o objetivo de verificar se o sistema acompanha a variação de temperatura lida. O sensor foi excitado com um ferro de solda quente, o qual foi posicionado próximo ao NTC do sistema de aquisição. O sistema de aquisição está no mesmo ambiente do dispositivo sensor. O resultado esperado é o gráfico de monitoramento seguindo a temperatura lida pelo sensor corretamente e o arquivo de dados com todas as aquisições realizadas durante o teste.

A Figura 61 mostra o resultado do teste do sensor de temperatura. Nela pode ser observado que o sensor realmente atingiu a

temperatura de aproximadamente 100°C conforme proposto para este teste e posteriormente retornou a temperatura ambiente. O sistema de aquisição adquiriu todos os pontos adequadamente, sem deixar a curva com nenhum tipo de descontinuidade, exceto no momento inicial, onde o sistema iniciou a aquisição dos dados. Neste momento verifica-se que o sinal passou de zero para aproximadamente 25°C que é a temperatura ambiente.

Figura 61: Resultados mostrados pela interface gráfica durante o teste do sensor de temperatura



Fonte: Produção do próprio autor

A Figura 62 mostra o arquivo de dados gerado durante o teste do sensor de temperatura. Nesta figura o arquivo de dados foi aberto no Excel e os dados adquiridos na figura mostram o início do teste, onde o sistema iniciou a aquisição do sensor de temperatura passando de zero para 27,7°C.

dispositivo 1, o dispositivo 0x0308, que equivale ao sensor de oxigenação do sangue, o dispositivo 2, o dispositivo 0x0302, que equivale ao sensor de temperatura corporal e o dispositivo 3, o dispositivo 0x0309, que equivale ao sensor de batimentos cardíacos.

Figura 63: Tela de identificação dos dispositivos do teste com os três sensores conectados a rede.

	Coordenador	Dispositivo 1	Dispositivo 2	Dispositivo 3
Network short address	0000	796F	7970	7971
IEEE Addr 64Bits	0050 C208 3408 F401	0050 C201 3002 080C	0050 C204 EC09 000F	0050 C213 8804 8C00
Router				
Logical Type	Coordenador	End Device	End Device	End Device
APS	40	40	40	40
MAC	8E	8C	8C	8C
MFCode	0210	0210	0210	0210
Buffer	50	50	50	50
Tx	0000	0000	0000	0000
EP	08	08	08	08
Profile ID	0104	0104	0104	0104
Devices	0007	0308	0302	0309
Ver	00	00	00	00
Flags	00	00	00	00

Fonte: Produção do próprio autor

A Figura 64 mostra a interface com o usuário e os gráficos dos dados adquiridos no teste com os três dispositivos sensores biomédicos conectados a rede. Pode ser percebido que os gráficos possuem leituras em faixas coerentes com o projeto do sistema de aquisição.

Figura 64: Interface com o usuário e gráficos do teste do sistema com os três dispositivos conectados a rede.



Fonte: Produção do próprio autor

Na Figura 65 é apresentado parte dos dados adquiridos e armazenados em arquivo texto durante o teste com os três dispositivos sensores conectados a rede sem fios *Zigbee*.

Figura 65: Imagem do arquivo de dados gravado durante o teste com os três dispositivos sensores conectados a rede.

Dados sistema de aquisição			
Data	Dispositivo 1	Dispositivo 2	Dispositivo 3
30/06/2013 23:03:37	95	35,4	70
30/06/2013 23:03:38	95	35,4	70
30/06/2013 23:03:39	95	35,4	70
30/06/2013 23:03:40	95	35,4	70
30/06/2013 23:03:41	95	35,3	70
30/06/2013 23:03:42	95	35,3	70
30/06/2013 23:03:43	95	35,3	70
30/06/2013 23:03:44	95	35,3	70
30/06/2013 23:03:45	95	35,3	70
30/06/2013 23:03:46	95	35,3	70
30/06/2013 23:03:47	95	35,3	70
30/06/2013 23:03:48	95	35,3	70
30/06/2013 23:03:49	96	35,3	70
30/06/2013 23:03:50	96	35,2	70
30/06/2013 23:03:51	96	35,2	70
30/06/2013 23:03:52	96	35,2	70
30/06/2013 23:03:53	96	35,2	70
30/06/2013 23:03:54	96	35,2	70
30/06/2013 23:03:55	96	35,2	70
30/06/2013 23:03:56	96	35,3	70
30/06/2013 23:03:57	96	35,3	70
30/06/2013 23:03:58	96	35,3	70
30/06/2013 23:03:59	96	35,3	70
30/06/2013 23:04:00	96	35,3	70
30/06/2013 23:04:01	96	35,4	70

Fonte: Produção do próprio autor

Pôde ser observado nos resultados mostrados acima, que o sistema completo com os três sensores propostos fazendo aquisições no mesmo momento, funcionou como o esperado em projeto. As leituras dos sinais apresentaram valores coerentes com os valores observados quando as medições foram realizadas de forma independente.

Os gráficos de dados foram atualizados da forma correta, assim como os arquivos de dados gerados, mostrando ao usuário do sistema os níveis de medição ao longo do tempo, conforme proposto neste desenvolvimento.

4.4.1 Testes do sistema com três voluntários

Este teste tem como objetivo avaliar o sistema em uso em três voluntários, adquirindo os sinais dos três sensores ao mesmo tempo e avaliando as diferenças entre as medições obtidas.

Os voluntários 1,2 e 3 são pessoas saudáveis sendo que o voluntário 1 é do sexo masculino e os voluntários 2 e 3 do sexo feminino com idades de 28, 25 e 65 anos respectivamente.

Cada um dos voluntários será monitorado durante cinco minutos em cada um dos sensores. Estes dados serão mostrados via

interface gráfica do *software* de monitoramento e os dados serão gravados em arquivo texto. Será utilizado um sensor comercial para medir a temperatura corporal, conforme mostrado na Figura 66, oxigenação do sangue e batimentos cardíacos, conforme mostra a Figura 67. Desta forma é possível a confecção de uma tabela resumo com a média dos sinais medidos pelo sistema de aquisição e pelo sensor comercial, sendo assim possível comparar as medições entre os voluntários e os dispositivos sensores.

Figura 66: Foto do sensor de temperatura comercial digital utilizado na medição da temperatura corporal.



Fonte: Produção do próprio autor.

Figura 67: Foto do sensor comercial utilizado para a medição da oxigenação no sangue e dos batimentos cardíacos. Modelo CMS50F.



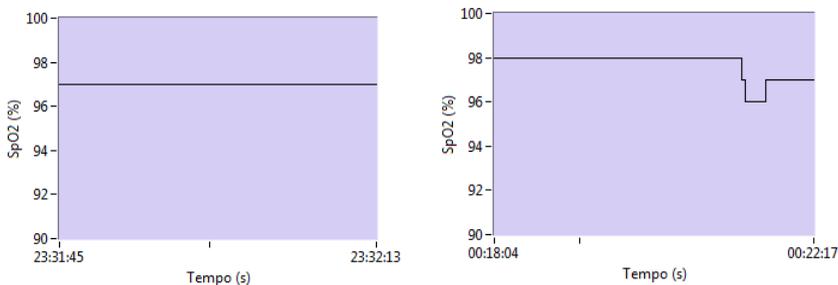
Fonte: Produção do próprio autor.

A metodologia utilizada nesta avaliação consiste em iniciar a rede *Zigbee*, conforme já mencionado nos testes anteriores, posicionar o sensor de temperatura na axila do voluntário 1, o sensor de oximetria no dedo indicador do voluntario 2 e o sensor de batimentos cardíacos também no dedo indicador do voluntário 3. O sistema de aquisição de

dados será posicionado no mesmo ambiente em que os voluntários estão a poucos metros dos mesmos.

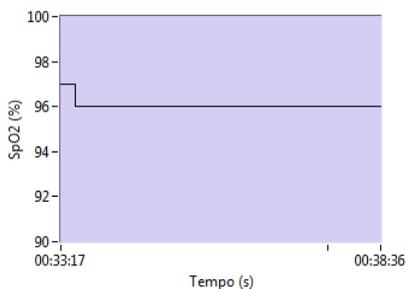
O gráfico de monitoramento da medição do sinal de oxigenação no sangue em cada um dos voluntários é mostrado na Figura 68.

Figura 68: Resultados mostrados pela interface gráfica durante o teste de medição da oxigenação do sangue nos três voluntários.



a) Voluntário 1

b) Voluntário 2

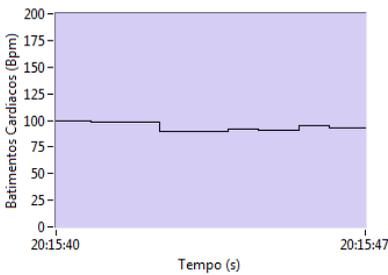


Voluntário 3

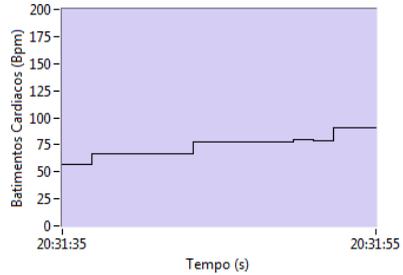
Fonte: produção do próprio autor.

O gráfico de monitoramento da medição do sinal de batimentos cardíacos em cada um dos voluntários é mostrado na Figura 69.

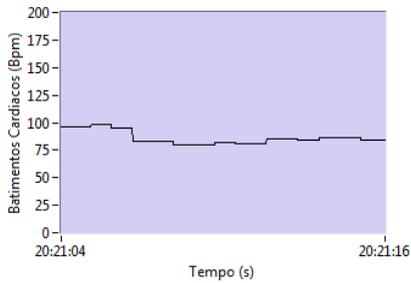
Figura 69: Resultados mostrados pela interface gráfica durante o teste de medição da frequência cardíaca nos três voluntários.



a) Voluntário 1



b) Voluntário 2

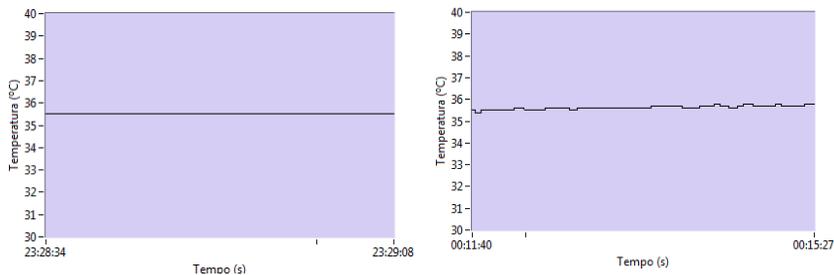


c) Voluntário 3

Fonte: produção do próprio autor.

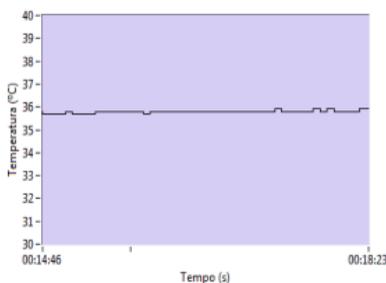
O gráfico de monitoramento da medição do sinal de temperatura corporal em cada um dos voluntários é mostrado na Figura 70.

Figura 70: Resultados mostrados pela interface gráfica durante o teste de medição da temperatura corporal nos três voluntários.



a) Voluntário 1

b) Voluntário 2



c) Voluntário 3

Fonte: produção do próprio autor.

Tabela 17: Média dos valores lidos no sensor de oxigenação do sangue nos testes com os três voluntários.

	Oxigenação (% SpO2)		
	Sensor da Rede	Sistema Comercial	Erro
Voluntário 1	97	97	0,0%
Voluntário 2	97	98	-1,0%
Voluntário 3	95	96	-1,0%
		Erro Médio	-0,7%

Fonte: produção do próprio autor.

Com base nos resultados dos testes com o oxímetro mostrados na Figura 68 e na Tabela 17, pode ser percebido que a medição do sensor da rede ficou muito próxima da medição do sistema comercial,

com erro médio de 0,7%. Nota-se também que este sinal não apresenta grande variação ao longo do tempo.

Para o sensor de batimentos cardíacos, os resultados são mostrados na Figura 69 e na Tabela 18. Nelas pode ser percebido que esta medição varia bastante com o tempo, sendo dependente do estado do paciente, mais baixo em repouso e mais rápido em exercício. O sinal do sensor implementado na rede ficou mais instável quando comparado com o sensor comercial, apresentando variações relativamente grandes durante alguns momentos da medição. O sinal encontrado variou para valores maiores que o sinal medido pelo sistema comercial em alguns instantes, e para valores menores que o medido no sistema comercial em outro, com erro médio de 1,7%, sendo este um erro randômico.

Tabela 18: Média dos valores lidos no sensor de batimentos cardíacos nos testes com os três voluntários.

	Batimentos (bpm)		
	Sensor da	Sistema	Erro
Voluntário 1	95	89	6,7%
Voluntário 2	75	78	-3,8%
Voluntário 3	87	85	2,4%
		Erro Médio	1,7%

Fonte: Produção do próprio autor.

Notou-se durante o teste que o sensor utilizado é muito sensível a luminosidade do ambiente, o que em alguns casos ocasiona leituras erradas. Para minimizar este problema recomenda-se a utilização de um material, como um velcro preto, ao redor do sensor para melhor fixação no dedo e também minimizar a entrada de luz externa no sensor.

Os testes de medição de temperatura corporal mostrados na Figura 70 e na Tabela 19, mostram que o sinal de temperatura varia lentamente e quando se compara a medição do sistema implementado com a medição do sistema comercial se observa um erro médio de 1,7%.

A medição do sistema comercial se mostrou em todos os casos um pouco maior que a medição do sistema implementado, com uma variação praticamente na mesma faixa. Deste fato pode ser concluído que o sistema implementado apresenta um erro sistêmico comparado ao sensor comercial utilizado.

Tabela 19: Média dos valores lidos no sensor de temperatura corporal nos testes com os três voluntários.

	Temperatura (°C)		
	Sensor da Rede	Sistema Comercial	Erro
Voluntário 1	35,5	36,2	-1,9%
Voluntário 2	35,7	36,3	-1,7%
Voluntário 3	35,8	36,3	-1,4%
		Erro Médio	-1,7%

Fonte: Produção do próprio autor.

CAPÍTULO 5 – DISCUSSÕES E CONCLUSÕES

O trabalho descrito nesta dissertação mostrou que a tecnologia de rede sem fios *Zigbee* pode ser utilizada com sucesso na transmissão de sinais bioelétricos, auxiliando na inserção de novas tecnologias na automação hospitalar e em tratamentos de pacientes que necessitam de cuidados em casa.

O alcance conseguido com o sistema implementado, apresentado no capítulo 2, pode ser considerado dentro do esperado na utilização deste protocolo, tanto em locais com obstáculos, quanto em locais livres. Esta tecnologia de rede sem fios possibilita, devido ao grande número de *clusters* e estruturas já desenvolvidas, uma implementação de rede relativamente simples, e com boa capacidade de ampliação dos números de dispositivos e aplicação.

O protocolo *Zigbee* pode ter em uma mesma rede, um elevado número de dispositivos, contudo em configurações de rede como a utilizada neste trabalho, onde se tem um dispositivo coordenador que funciona como concentrador de dados, observam-se algumas limitações quanto ao envio e recebimento de mensagens. Estas limitações se devem a alguns fatores:

- 1- Memória necessária para armazenar todos os endereços dos nós da rede para envio e recebimento de mensagens: Este item não foi observado em nenhum teste neste trabalho devido ao fato desta implementação ter sido feita com um baixo número de dispositivos. Foram utilizados quatro nós *Zigbee*, um coordenador e três dispositivos fins (sensores).
- 2- Banda da rede *Zigbee*: A banda da rede pode ser influenciada por diversos fatores que limitam o desempenho final da rede. Uma rede *Zigbee* tem uma banda máxima de 250kbps, a qual é compartilhada com todos os dispositivos vizinhos na rede. Na prática, a banda possível não chega a este valor máximo, e deve ser considerado um valor de banda dependente de interferências (podem ocorrer interferências com o protocolo Wi-Fi), densidade da rede, o que limita o número de mensagens da aplicação por segundo, número de tentativas de retransmissão de pacotes de dados e padrões de comunicação. Novamente, devido ao fato da rede implementada ser pequena, apenas com quatro dispositivos (um coordenador da rede e três sensores), e

estar em um ambiente controlado de laboratório, a limitação de banda não foi observada.

Na comunicação entre o nó concentrador de dados e o computador também se pode observar potenciais limitações devido a banda da comunicação serial (no *kit* de desenvolvimento existe um circuito integrado que converte o sinal USB para serial), realizada entre o computador e o microcontrolador. Esta limitação depende da velocidade de comunicação escolhida para a aplicação e da quantidade de bits necessários para a troca de mensagens no processo de atualização dos dados.

Nesta aplicação utilizou-se uma velocidade de 38400 bps (bits por segundo), e em média são trocados 150 *bytes* entre o computador e o microcontrolador para se realizar a atualização de uma variável. Com isso verificou-se que é possível a atualização do valor de aproximadamente 30 variáveis a cada segundo. A utilização de apenas três nós sensores, com grandezas medidas com taxas de aquisição relativamente pequenas não gerou restrições a rede implementada, contudo caso seja necessário aumentar a taxa de aquisição de algum sinal, ou aumentar o número de nós na rede, deve-se aumentar a velocidade da comunicação serial.

A utilização de sistemas de redes sem fios para a aquisição e transmissão de sinais bioelétricos proporciona aos pacientes maior comodidade no seu dia a dia, e aos hospitais e clínicas uma maior facilidade no que diz respeito a infraestrutura.

O sistema de aquisição, armazenamento e tratamento de dados desenvolvido em Labview se mostrou muito flexível, capaz de identificar diversos parâmetros dos dispositivos conectados a rede, como endereço de rede, número do dispositivo e aplicação disponível nele, proporcionando ao usuário do sistema facilidade de uso.

Quanto ao tratamento dos dados, pôde-se observar que o usuário tem acesso as grandezas medidas de forma visual, podendo armazenar os dados e realizar verificações não apenas pontuais, como comumente é feito na medicina atual, mas sim, ter acesso a um histórico de dados podendo observar de forma contínua o estado do paciente, e verificando se em algum momento ocorreu algum tipo de anomalia.

O foco deste trabalho não foi o desenvolvimento de sensores biomédicos, mas sim a integração destes sensores em um sistema de comunicação sem fios utilizando o protocolo *Zigbee*. Sob esta ótica, o trabalho possibilitou diferentes formas de integração de dispositivos a

rede, um dispositivo através de comunicação serial (sensor de oximetria), dois dispositivos através da leitura de um sinal analógico (sensor de batimentos cardíacos e sensor de temperatura). Isto mostra flexibilidade na adoção desta tecnologia nos dias de hoje, tornando possível a adaptação, através de diferentes formas, dos sistemas biomédicos existentes e em uso hoje, com o sistema de comunicação sem fios *Zigbee* e não requerendo em vários casos a substituição do equipamento existente por outro já com a tecnologia sem fios inserida.

Como conclusão geral deste trabalho, pode-se verificar que iniciativas que facilitam a entrada da medicina no dia a dia das pessoas geram uma quebra de paradigma da medicina tradicional, direcionada por um modelo de eventos, tal qual, ir ao médico quando se está doente, ou fazer uma avaliação de *check up* uma vez ao ano, para um modelo onde se pode monitorar continuamente a saúde e o bem estar das pessoas através de bio sensores e outras tecnologias mais modernas aliadas a redes de informações.

Este modelo possibilita que as pessoas usuárias desta nova tecnologia sejam mais proativas na manutenção da saúde, assim como permite que os profissionais ligados a área da saúde tomem decisões baseados em mais informações, com dados precisos e com base de comparação. Possibilitando, inclusive a avaliação de dados históricos, mostrando a evolução ou regressão de determinado tipo de doença.

Como melhorias e ideias complementares a este trabalho, acredito ser necessário adicionar ao sistema mais dispositivos, visto que neste trabalho limitou-se a uma rede com quatro dispositivos, assim como outros tipos de sensores, além de dispositivos atuadores, os quais podem tomar ações específicas com base nos dados recebidos e interpretados, como no caso de reguladores de remédios, ou respiradores, ou mesmo alarmes sonoros.

Pode-se também adicionar a camada de segurança a rede *Zigbee*, proporcionando restrição de acesso aos dados a dispositivos específicos, adicionar a função de baixo consumo de energia aos dispositivos sensores, otimizando o uso de baterias, e também utilizar outra topologia de rede, como por exemplo, a topologia MESH, a qual possibilita boa flexibilidade para uma rede com maior número de dispositivos.

REFERENCIAS BIBLIOGRÁFICAS

ADAMS, Jon T. (2009), **An Introduction to IEEE STD 802.15.4**. Freescale Semiconductors.

ANDRIGHETTO, Eduardo. **Sistema de Processamento de Sinais Biomédicos: Rede Wireless Zigbee com Aplicação do Padrão IEEE802.15.4**. Dissertação de Mestrado. UFSC, 2008.

BAILEY, Joseph. **Wireless Pulse Oximeter**. Worcester Polytechnic Institute. 2008.

BRANQUINHO, O C.; REGGIANI, N; ANDREOLLO, A. G. (2005). **Redes de comunicação de dados sem fio – uma análise de desempenho**. ISA SHOW SOUTH AMERICA Feira Sul Americana e 5º Congresso Internacional de Automação, Sistemas e Instrumentação. São Paulo; [s.n.].

CAMPOS DO PRADO, Rafael. **Uma Proposta de Sensoriamento Remoto de Temperatura para Controle de Condicionadores de Ar Utilizando o Padrão IEEE 802.15.4**. Dissertação (Mestrado em Automação Industrial) – Universidade do Estado de Santa Catarina, Joinville, 2008.

CARRARA, Dirceu. **Oximetria de Pulso Arterial**. Conselho Regional de Enfermagem de São Paulo. 2009.

CHÁVEZ, Dr. Ignacio. HERRERA, Dr. Eulo Lupi. **Programa De Actualización Continua Para El Cardiólogo. PAC Cardio 2 – Tomo 10**. Disponível em: <http://www.drscope.com/pac/cardio-2/10/index.htm> Acessado em Janeiro/2013.

LEITE, Cicília R.M., Bruno G. de Araujo, Ricardo A. De M. Valentim, Gláucio B. Brandão e Ana M. G. Gueirreiro. **Novas Tecnologias para Automação Hospitalar**, 2010.

COSTA FILHO, Antonio Agripino da (2011). **Sistema de Aquisição e Tratamento de Dados Usando o Padrão IEEE 802.15.4 de Redes de Sensores Sem Fio**. COBENGE – Congresso Brasileiro de Educação em

Engenharia, Blumenau.

DAINTREE Networks, **Getting Started with Zigbee and IEEE 802.15.4.** 2008. Disponível em http://www.daintree.net/downloads/whitepapers/zigbee_primer.pdf em 16 outubro de 2012.

DBC Educacional, **Como funciona um Oxímetro de Pulso Digital.** Disponível em http://www.oxigenio.com/guia-dos-oximetros-de-pulso-digitais-para-oxigenioterapia-tipo-dedal/como_funciona_um_pulso_oximetro_digital.htm em 16 fevereiro de 2013.

ESTEVEES, Tiago. **Cuidados Básicos de Saúde – Sinais Vitais.** Notas de aula. Koogan, 2010.

FARAHANI, Shahin. **Zigbee Wireless Networks and Transceivers.** Elsevier. 2008.

FREESCALE SEMICONDUCTOR. 13213EVKUG, **13213 Evaluation Kits User`s Guide** Rev 1.1. 2007

FREESCALE SEMICONDUCTOR. ZTCRM, **Freescale Zigbee™ Test Client Reference Manual.** Rev 0.0. 2007.

FREESCALE SEMICONDUCTOR. BSBBZTCRM, **BeeStack™ BlackBox Zigbee™ Test Client (ZTC).** Rev 1.2. 2011.

FREESCALE SEMICONDUCTOR. TTUG, **Freescale Test Tool User Guide.** Rev 1.6. 2011.

FREESCALE SEMICONDUCTOR. ZAUG, **Freescale Zigbee Application User Guide.** Rev 1.3. 2008.

FREESCALE SEMICONDUCTOR. BSSRM, **Freescale BeeStack™ Software Reference Manual.** Rev 1.0. 2007.

FREESCALE SEMICONDUCTOR. BSADG, **Freescale BeeStack™ Application Development Guide.** Rev 1.1. 2008.

FREESCALE SEMICONDUCTOR. FSPRM, **Freescale Platform Reference Manual**. Rev 1.0. Freescale. 2008.

FREESCALE SEMICONDUCTOR. ZCLRM, **Freescale Zigbee™ Cluster Library (ZCL) Reference Manual**. Rev 1.2. 2008.

FREESCALE SEMICONDUCTOR. ZEDESUG, **Zigbee™ environment Demonstration (ZeD) Embedded Software Design User's Guide**. Rev 1.3. 2008.

FREESCALE SEMICONDUCTOR. ZEDSUG, **Zigbee™ environment Demonstration (ZeD) Software User's Guide**. Rev 1.2. 2008.

FREESCALE SEMICONDUCTOR. BSDO, **Freescale BeeStack™ Documentation Overview**. Rev 1.0. 2008.

FREESCALE SEMICONDUCTOR. BKWCTKUG, **BeeKit Wireless Connectivity Toolkit User's Guide**. Rev 2.1. 2011.

FREESCALE SEMICONDUCTOR. MC1321x, **MC13211/212/213 Zigbee™- Compliant Platform - 2.4 GHz Low Power Transceiver for the IEEE® 802.15.4 Standard plus Microcontroller**. Rev 1.8. Freescale Semiconductor Technical Data. 2009.

FREESCALE SEMICONDUCTOR. MC1321xRM, **MC13211/212/213 Zigbee™- Compliant Platform 2.4 GHz Low Power Transceiver for the IEEE® 802.15.4 Standard plus Microcontroller Reference Manual**. Rev 1.6. Freescale Semiconductor Technical Data. 2010.

FREESCALE SEMICONDUCTOR. MC1321xEVKRM, **MC1321x Evaluation Kit (EVK) Reference Manual**. Rev 1.2. 2008.

GALEEV, Mikhail. **Home networking with Zigbee**. Artigo EETimes. 2004. Disponível em www.eetimes.com/design/embedded/4006430/Home-networking-with-Zigbee em 23 de junho de 2012.

GISLASON, Drew. **Zigbee Wireless Networking**. Newnes. 2008.

IEEE Std802.15.4. IEEE Computer Society, Part15.4: Wireless

Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANS). Published by The Institute of Electrical and Electronics Engineers, Inc. New York, USA, 1 October 2003.

ISTEPANIAN, R. S. H.; WOODWARD, B.; 1997. **Microcontroller-Based Underwater Acoustic ECG Telemetry System**. IEEE Transactions on Information Technology in Biomedicine, v. 1, n. 2 (Jun.), p. 150-154.

LOPEZ, Santiago. **AN4327 - Pulse Oximeter Fundamentals and Design**. Rev 2. Freescale. 2012.

MARTINCOSKI, Daniel Henrique. **Sistema Para Telemetria De Eletrocardiograma Utilizando Tecnologia Bluetooth**. Dissertação de Mestrado. UFSC, 2003.

MONSIGNORE, Ferdinando. **Sensoriamento de Ambiente Utilizando o Padrão Zigbee**. Dissertação de Mestrado. UFSCAR, 2007.

NORRIS, M. (2005). **Single-chip Zigbee for indoor mobile telemetry**. The IEEE Seminar on Telemetry and Telematics, p.10/1-10/4.

PARK, J.; SON, J.; SEO, H.; ISHIDA, M.; 1998. **4-subject 4-channel optical telemetry system for use in electrocardiograms**. In: IEEE INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS AND SYSTEMS (5. : Sep. 1998 : Lisboa). Proceedings. Lisboa. v. 3, p. 251-254.

PIANCINI, Charles R. S. **Sistema de monitoramento de temperatura corporal**. UnicemP. 2005.

POTTER, A. P. PERRI, A. G. **Fundamentos de enfermagem: conceitos, processo e pratica**. 5ªed. Rio de Janeiro; Guanabara Koogan, 2004.

PORTO, Celmo C. **Exame Clínico: Bases para a prática médica**. 6 ed, Rio de Janeiro. Guanabara Koogan, 2008.

PUTTINI, Ricardo Staciarini. **Tecnologias de Redes MANs e WANs**.

Material de Aula. Universidade de Brasília, 2000.

QURESHI, Hamzah. **Design of a Wireless Pulse Oximeter for use in a Clinical Diagnostic System**. McMaster University, Hamilton, Ontario, Canada. Abril, 2009

SMITH, Joseph M., **Wireless Health Care**, IEEE Spectrum, 2011. [Acessado em 20 Dezembro de 2012] Disponível em <http://spectrum.ieee.org/biomedical/devices/wireless-health-care>

STEINHAUSER, Paulo Luis. (2012), **Ziglar - Utilização de redes sem fio Zigbee para Acessibilidade aos portadores de deficiência física**. UNIDAVI.

SUNROM TECHNOLOGIES. 1157, **Heart Beat Sensor Datasheet**. Disponível em www.sunrom.com/p-556.html em março de 2013.

TETZLAFF AAS (Hi Technologies). **Bases de física para Oximetria**. [online] 2010 Abr. [acessado em 18, Outubro de 2012]. Disponível em: <http://hitechnologies.com.br/artigos/conhecimentos-basicos/bases-de-fisica-para-oximetria/>

VALENTIM, R. A. M. ; MORAIS, A. H. F. ; SOUZA, V. S. V ; ARAUJO JUNIOR, H. B. ; BRANDAO, G. B. ; GUERREIRO, A. M. G. . **Rede de Controle em Ambiente Hospitalar: um protocolo multiciclos para automação hospitalar sobre IEEE 802.3 com IGMP Snooping**. Revista Ciência e Tecnologia, v. 11, p. 19, 2009

VASQUES, Bruna Luisa Ramos Prado. 2010. Disponível em http://www.gta.ufrj.br/grad/10_1/zigbee/referencias.html visitada em setembro/2012.

WATANABE, Ana T Y. **Monitoramento, controle e parametrização de eletrodomésticos numa rede sem fio via interface virtual remota**. Dissertação (Mestrado em Automação Industrial) – Universidade do Estado de Santa Catarina, Joinville, 2008.

WEBSTER, J. G. (2010). **Medical Instrumentation Application and Design**, Wiley.

WOODWARD, Stephen W. **Build your own optical heart-rate sensor.** Disponível em: electronicdesign.com/article/components/build-your-own-optical-heart-rate-sensor6343 Acessado em Dezembro/2012.

ZIGBEE ALLIANCE. *Zigbee Specification*, ZigBee Document 053474r17. 2008.

ZIGBEE ALLIANCE. *Zigbee Home Automation Public Application Profile*, *Zigbee* Document 053520r26. Revisão 26. Versão 1.1. 2010.

ZIGBEE ALLIANCE. *Zigbee Cluster Library Specification*, ZigBee Document 075123r02ZB. 2008.

ZIGBEE ALLIANCE. *Zigbee Health Care Profile Specification*, ZigBee Document 075360r15. Revisão 15 Versão 1.0. 2010.

APÊNDICE – LINK QUALITY INDICATOR (LQI)

De acordo com FARAHANI (2008), o indicador de qualidade de conexão, ou *link quality indicator* (LQI) é uma indicação da qualidade dos pacotes de dados recebidos pelo dispositivo receptor da mensagem. A força do sinal recebido, ou *received signal strength* (RSS) pode ser utilizada como uma medida de qualidade do sinal. A RSS é a medida da energia total do sinal recebido. Outra forma de avaliar a qualidade do sinal é através da razão entre a energia do sinal desejada e a energia do ruído total na banda, ou seja a relação sinal ruído, também chamada de *signal to noise ratio* (SNR).

De forma geral um sinal com uma relação sinal ruído maior é considerado como com uma qualidade maior. A qualidade de conexão pode ser avaliada utilizando-se as duas medidas de forma conjunta, a energia do sinal e a relação sinal ruído.

A medição da indicação da qualidade do sinal é realizada para cada pacote de mensagem recebida. O LQI deve ter pelo menos oito níveis únicos e é reportado para a camada MAC, a qual o disponibiliza para as camadas de rede e de aplicação para qualquer tipo de análise. Por exemplo, a camada de rede pode utilizar os níveis de LQI dos dispositivos para decidir qual o melhor caminho para o roteamento de mensagens na rede. De forma geral o caminho com o maior LQI tem uma maior chance de que a mensagem seja entregue ao seu destino.

ANEXO – COMANDOS DO ZIGBEE TEST CLIENT (ZTC)

O protocolo Zigbee e a Freescale disponibilizam uma série de comandos e eventos que podem ser enviados e processados através de troca de mensagens utilizando o ZTC. A Tabela 20 abaixo apresentam alguns destes comandos, deixando claro quais são o Opcode Group e o Opcode de cada comando e descrevendo o nome de cada um destes. Neste trabalho nem todos estes comandos foram utilizados. Maiores informações sobre a descrição de cada comando pode ser encontrado em FREESCALE SEMICONDUCTOR ZTCRM, 2007 e BSBZTCRM, 2011.

Tabela 20: Comandos do ZTC

TYPE	OpCode Group	OpCode	NAME
COMMAND	96	70	NLME-ROUTE-DISCOVERY.Request
COMMAND	00	10	PowerOn.Request
COMMAND	69	00	TransmitCountedPackets.Req
COMMAND	69	01	ResetPacketCount.Req
COMMAND	69	02	RetrievePacketCount.Req
COMMAND	69	04	BufferTest.Req
COMMAND	69	05	BufferTestGroupReq
COMMAND	69	08	RouteDiscoveryReq
COMMAND	69	0A	FreeformReq
COMMAND	69	0D	BroadcastReq
COMMAND	85	00	MacAssociate.Request
COMMAND	85	01	MacAssociate.Response
COMMAND	85	02	MacDisassociate.Request
COMMAND	85	03	MacGetPIBAttribute.Request
COMMAND	85	05	MacOrphan.Response
COMMAND	85	06	MacReset.Request
COMMAND	85	07	MacRxEnable.Request
COMMAND	85	08	MacScan.Request
COMMAND	85	09	MacSetPIBAttribute.Request
COMMAND	85	0A	MacStart.Request
COMMAND	87	00	MacData.Request
COMMAND	87	01	MacPurge.Request
COMMAND	95	00	AspGetTime.Request
COMMAND	95	01	AspGetInactiveTime.Request
COMMAND	95	02	AspDoze.Request
COMMAND	95	03	AspAutoDoze.Request
COMMAND	95	04	AspHibernate.Request
COMMAND	95	05	AspWake.Request
COMMAND	95	06	AspEvent.Request
COMMAND	95	07	AspTrim.Request
COMMAND	95	08	AspDdr.Request

TYPE	OpCode Group	OpCode	NAME
COMMAND	95	09	AspPort.Request
COMMAND	95	0A	AspClko.Request
COMMAND	96	34	NLME-NETWORK-DISCOVERY.Request
COMMAND	96	35	NLME-NETWORK-FORMATION.Request
COMMAND	96	36	NLME-PERMIT-JOINING.Request
COMMAND	96	37	NLME-START-ROUTER.Request
COMMAND	96	38	NLME-JOIN.Request
COMMAND	96	39	NLME-DIRECT-JOIN.Request
COMMAND	96	3A	NLME-LEAVE.Request
COMMAND	96	3B	NLME-RESET.Request
COMMAND	96	3C	NLME-SYNC.Request
COMMAND	96	3D	NLME-GET.Request
COMMAND	96	3E	NLME-SET.Request
COMMAND	99	00	APSME-BIND.Request
COMMAND	99	03	APSME-GET.Request
COMMAND	99	06	APSME-SET.Request
COMMAND	99	09	APSME-UNBIND.Request
COMMAND	99	0B	APSME-RESET.Request
COMMAND	99	CF	APSME-ESTABLISH-KEY.Request
COMMAND	99	D0	APSME-ESTABLISH-KEY.response
COMMAND	99	D1	APSME-TRANSPORT-KEY.Request
COMMAND	99	D3	APSME-REQUEST-KEY.Request
COMMAND	99	D4	APSME-SWITCH-KEY.Request
COMMAND	99	E8	APSME-ADD-GROUP.Request
COMMAND	99	E9	APSME-REMOVE-GROUP.Request
COMMAND	99	EA	APSME-REMOVE-ALL-GROUPS.Request
COMMAND	99	F3	APSME-UPDATE-DEVICE.Request
COMMAND	99	F4	APSME-REMOVE-DEVICE.Request
COMMAND	9A	33	NLDE-DATA.Request
COMMAND	9C	00	APSDE-DATA.Request
COMMAND	9F	00	FindNumberOfEndPoint
COMMAND	9F	00	Register-EndPoint
COMMAND	9F	01	DeRegister-EndPoint
COMMAND	9F	01	FindEndPointDescription
COMMAND	9F	02	AFDE-ZCL-DATA.Request
COMMAND	9F	02	FindEndPointIdList
COMMAND	9F	03	AFDE-MSG-DATA.Request
COMMAND	9F	06	AFDE-MSG-DATA-Aggregation.Request
COMMAND	A0	D0	ZDP-APSME-ESTABLISH-KEY.response
COMMAND	A2	00	ZDP-NWK_addr_req.Request
COMMAND	A2	01	ZDP-IEEE_addr_req.Request
COMMAND	A2	02	ZDP-NodeDescriptor.Request
COMMAND	A2	03	ZDP-PowerDescriptor.Request
COMMAND	A2	04	ZDP-SimpleDescriptor.Request
COMMAND	A2	05	ZDP-Active_EP_req.Request
COMMAND	A2	06	ZDP-Match_Desc_req.Request

TYPE	OpCode Group	OpCode	NAME
COMMAND	A2	10	ZDP-Complex_Desc_req
COMMAND	A2	11	ZDP-User_Desc_req.Request
COMMAND	A2	12	ZDP-Discovery_Cache.request
COMMAND	A2	13	ZDP-EndDeviceAnnounce.Request
COMMAND	A2	14	ZDP-User_Desc_set
COMMAND	A2	15	ZDP-System_Server_Discovery.Request
COMMAND	A2	16	ZDP-Discovery_Store.Request
COMMAND	A2	17	ZDP-Node_Desc_Store.Request
COMMAND	A2	18	ZDP-Power_Desc_Store.Request
COMMAND	A2	19	ZDP-Active_EP_Store.Request
COMMAND	A2	1A	ZDP-Simple_Desc_Store.Request
COMMAND	A2	1B	ZDP-Remove_Node_Cache.Request
COMMAND	A2	1C	ZDP-Find_Node_Cache.Request
COMMAND	A2	20	ZDP-End_Device_Bind.Request
COMMAND	A2	21	ZDP-BIND.Request
COMMAND	A2	22	ZDP-UNBIND.Request
COMMAND	A2	23	ZDP-Bind_Register.Request
COMMAND	A2	24	ZDP-Replace_Device.Request
COMMAND	A2	25	ZDP-Store_Bkup_Bind_entry.Request
COMMAND	A2	26	ZDP-Remove_Bkup_Bind_Entry.Request
COMMAND	A2	27	ZDP-Backup_Bind_Table.Request
COMMAND	A2	28	ZDP-Recover_Bind_Table.Request
COMMAND	A2	29	ZDP-Backup_Source_Bind.Request
COMMAND	A2	2A	ZDP-Recover_Source_Bind.Request
COMMAND	A2	30	ZDP-Mgmt_Nwk_Disc.Request
COMMAND	A2	31	ZDP-Mgmt_Lqi.Request
COMMAND	A2	32	ZDP-Mgmt_Rtg.Request
COMMAND	A2	33	ZDP-Mgmt_Bind.Request
COMMAND	A2	34	ZDP-Mgmt_Leave.Request
COMMAND	A2	35	ZDP-Mgmt_Direct_Join.Request
COMMAND	A2	36	ZDP-Mgmt_Permit_Joining.Request
COMMAND	A2	36	ZDP-NLME-PERMIT-JOINING.Request
COMMAND	A2	37	ZDP-Mgmt_Cache.Request
COMMAND	A2	39	ZDP-NLME-DIRECT-JOIN.Request
COMMAND	A2	3C	ZDP-NLME-SYNC.Request
COMMAND	A2	50	ZDP-NLME-NETWORK-DISCOVERY.Request
COMMAND	A2	CF	ZDP-APSME-ESTABLISH-KEY.Request
COMMAND	A2	D1	ZDP-APSME-TRANSPORT-KEY.Request
COMMAND	A2	D2	ReadExtAddr.Request
COMMAND	A2	D3	ZDP-APSME-REQUEST-KEY.Request
COMMAND	A2	D4	ZDP-APSME-SWITCH-KEY.Request
COMMAND	A2	DA	ConfigurationAttributes.Init
COMMAND	A2	DC	StopNwk.Request
COMMAND	A2	E0	RestartNwk.Request
COMMAND	A2	E4	BeeStackSave.Request
COMMAND	A2	F3	ZDP-APSME-UPDATE-DEVICE.Request

TYPE	OpCode Group	OpCode	NAME
COMMAND	A2	F4	ZDP-APSME-REMOVE-DEVICE.Request
COMMAND	A2	FA	SetApsSecurityMaterials.Request
COMMAND	A2	FC	SetNwkSecurityMaterials.Request
COMMAND	A2	FE	Check-NeighborTable-ForRouting.Request
COMMAND	A3	00	ZTCModeSelect.request
COMMAND	A3	02	ZTCGetMode.request
COMMAND	A3	05	ZTC_AFRReset.request
COMMAND	A3	06	ZTC_APSReset.request
COMMAND	A3	07	ZTC_SetAPSReady.request
COMMAND	A3	08	ZTC_CPU_Reset.request
COMMAND	A3	DB	WriteExtAddr.Request
COMMAND	A3	DF	StartNwk.Request
EVENT	96	70	NLME-ROUTE-DISCOVERY.Request
EVENT	97	70	NLME-ROUTE-DISCOVERY.confirm
EVENT	00	00	NLME-ROUTE-ERROR.Indication
EVENT	A0	A1	ZDP-Bind.response
EVENT	00	10	PowerOn.Request
EVENT	69	00	TransmitCountedPackets.confirm
EVENT	69	01	ResetPacketCount.confirm
EVENT	69	02	RetrievePacketCount.confirm
EVENT	69	03	RetrievePacketCount.response
EVENT	69	04	BufferTest.confirm
EVENT	69	05	BufferTestGroup.confirm
EVENT	69	06	BufferTest.response
EVENT	69	07	BufferTestGroup.response
EVENT	69	08	RouteDiscovery.confirm
EVENT	69	09	RouteDiscovery.response
EVENT	69	0A	Freeform.confirm
EVENT	69	0B	Freeform.response
EVENT	69	0C	FreeformNoData.response
EVENT	69	0D	Broadcast.confirm
EVENT	69	0E	BroadcastAllDevices.Response
EVENT	69	0F	BroadcastRxOnIdle.Response
EVENT	69	10	BroadcastRouters.Response
EVENT	84	00	MacAssociate.Indication
EVENT	84	01	MacAssociate.Confirm
EVENT	84	02	MacDisassociate.Indication
EVENT	84	03	MacDisassociate.Confirm
EVENT	84	04	MacBeaconNotify.Indication
EVENT	84	05	MacGetPIBAttribute.Confirm
EVENT	84	06	MacGTS.Indication
EVENT	84	07	MacGTS.Confirm
EVENT	84	08	MacOrphan.Indication
EVENT	84	09	MacReset.Confirm
EVENT	84	0A	MacRxEnable.Confirm
EVENT	84	0B	MacScan.Confirm

TYPE	OpCode Group	OpCode	NAME
EVENT	84	0C	MacCommStatus.Indication
EVENT	84	0D	MacSetPIBAttribute.Confirm
EVENT	84	0E	MacStart.Confirm
EVENT	84	0F	MacSyncLoss.Indication
EVENT	84	10	MacPoll.Confirm
EVENT	84	11	MacInputError.Confirm
EVENT	86	00	MacData.Confirm
EVENT	86	01	MacData.Indication
EVENT	86	02	MacPurge.Confirm
EVENT	86	03	Promiscuous
EVENT	87	00	MacData.Request
EVENT	87	00	MacData.Request
EVENT	94	00	AspGetTime.Confirm
EVENT	94	01	AspGetInactiveTime.Confirm
EVENT	94	02	AspDoze.Confirm
EVENT	94	03	AspAutoDoze.Confirm
EVENT	94	04	AspHibernate.Confirm
EVENT	94	05	AspWake.Confirm
EVENT	94	06	AspEvent.Confirm
EVENT	94	07	AspTrim.Confirm
EVENT	94	08	AspDdr.Confirm
EVENT	94	09	AspPort.Confirm
EVENT	94	0A	AspClko.Confirm
EVENT	94	0B	AspTemp.Confirm
EVENT	94	0C	AspWake.Indication
EVENT	94	0D	AspIdle.Indication
EVENT	94	0E	AspInactive.Indication
EVENT	94	0F	AspEvent.Indication
EVENT	94	10	AspNvRam.Confirm
EVENT	94	11	AspBattery.Confirm
EVENT	95	00	AspGetTime.Request
EVENT	95	01	AspGetInactiveTime.Request
EVENT	95	02	AspDoze.Request
EVENT	95	03	AspAutoDoze.Request
EVENT	95	04	AspHibernate.Request
EVENT	95	05	AspWake.Request
EVENT	95	06	AspEvent.Request
EVENT	95	07	AspTrim.Request
EVENT	95	08	AspDdr.Request
EVENT	95	09	AspPort.Request
EVENT	95	0A	AspClko.Request
EVENT	96	34	NLME-NETWORK-DISCOVERY.Request
EVENT	96	35	NLME-NETWORK-FORMATION.Request
EVENT	96	36	NLME-PERMIT-JOINING.Request
EVENT	96	37	NLME-START-ROUTER.Request
EVENT	96	38	NLME-JOIN.Request

TYPE	OpCode Group	OpCode	NAME
EVENT	96	39	NLME-DIRECT-JOIN.Request
EVENT	96	3A	NLME-LEAVE.Request
EVENT	96	3B	NLME-RESET.Request
EVENT	96	3C	NLME-SYNC.Request
EVENT	96	3D	NLME-GET.Request
EVENT	96	3E	NLME-SET.Request
EVENT	97	41	NLME-NETWORK-DISCOVERY.Confirm
EVENT	97	42	NLME-NETWORK-FORMATION.Confirm
EVENT	97	43	NLME-PERMIT-JOINING.Confirm
EVENT	97	44	NLME-START-ROUTER.Confirm
EVENT	97	45	NLME-JOIN.Indication
EVENT	97	46	NLME-JOIN.Confirm
EVENT	97	47	NLME-DIRECT-JOIN.Confirm
EVENT	97	48	NLME-LEAVE.Confirm
EVENT	97	49	NLME-LEAVE.Indication
EVENT	97	4A	NLME-RESET.Confirm
EVENT	97	4B	NLME-SYNC.Indication
EVENT	97	4C	NLME-SYNC.Confirm
EVENT	97	4D	NLME-GET.Confirm
EVENT	97	4E	NLME-SET-NEIGHBOR-TABLE.Confirm
EVENT	97	4E	NLME-SET-nwkSecurityMaterialSet.Confirm
EVENT	97	4E	NLME-SET.Confirm
EVENT	98	00	APSME-ESTABLISH-KEY.Indication
EVENT	98	02	APSME-REMOVE-DEVICE.Indication
EVENT	98	03	APSME-REQUEST-KEY.Indication
EVENT	98	04	APSME-SWITCH-KEY.Indication
EVENT	98	05	APSME-TRANSPORT-KEY.Indication
EVENT	98	06	APSME-UPDATE-DEVICE.Indication
EVENT	98	06	ZDotoApplication.indication
EVENT	98	07	APSME-BIND.Confirm
EVENT	98	08	APSME-UNBIND.Confirm
EVENT	98	09	APSME-GET.Confirm
EVENT	98	0A	APSME-SET.Confirm
EVENT	98	0B	APSME-RESET.Confirm
EVENT	98	CF	APSME-ESTABLISH-KEY.Confirm
EVENT	98	CF	APSME-ESTABLISH-KEY.Confirm
EVENT	98	D1	APSME-TRANSPORT-KEY.Confirm
EVENT	98	D3	APSME-REQUEST-KEY.Confirm
EVENT	98	D4	APSME-SWITCH-KEY.Confirm
EVENT	98	E8	APSME-ADD-GROUP.Confirm
EVENT	98	E9	APSME-REMOVE-GROUP.Confirm
EVENT	98	EA	APSME-REMOVE-ALL-GROUPS.Confirm
EVENT	98	F3	APSME-UPDATE-DEVICE.Confirm
EVENT	98	F4	APSME-REMOVE-DEVICE.Confirm
EVENT	99	00	APSME-BIND.Request
EVENT	99	03	APSME-GET.Request

TYPE	OpCode Group	OpCode	NAME
EVENT	99	06	APSME-SET.Request
EVENT	99	09	APSME-UNBIND.Request
EVENT	99	0B	APSME-RESET.Request
EVENT	99	CF	APSME-ESTABLISH-KEY.Request
EVENT	99	D0	APSME-ESTABLISH-KEY.response
EVENT	99	D1	APSME-TRANSPORT-KEY.Request
EVENT	99	D3	APSME-REQUEST-KEY.Request
EVENT	99	D4	APSME-SWITCH-KEY.Request
EVENT	99	E8	APSME-ADD-GROUP.Request
EVENT	99	E9	APSME-REMOVE-GROUP.Request
EVENT	99	EA	APSME-REMOVE-ALL-GROUPS.Request
EVENT	99	F3	APSME-UPDATE-DEVICE.Request
EVENT	99	F4	APSME-REMOVE-DEVICE.Request
EVENT	9A	33	NLDE-DATA.Request
EVENT	9B	3F	NLDE-DATA.Confirm
EVENT	9B	40	NLDE-DATA.Indication
EVENT	9C	00	APSDE-DATA.Request
EVENT	9D	00	APSDE-DATA.Confirm
EVENT	9D	01	APSDE-DATA.Indication
EVENT	9E	00	Register.Confirmation
EVENT	9E	01	Deregister.Confirmation
EVENT	9E	02	AFDE-DATA.Confirmation
EVENT	9E	03	AFDE-KVP-DATA.Indication
EVENT	9E	04	AFDE-MSG-DATA.Indication
EVENT	9F	00	FindNumberOfEndPoints
EVENT	9F	00	Register-EndPoint
EVENT	9F	01	DeRegister-EndPoint
EVENT	9F	01	FindEndPointDescription
EVENT	9F	02	FindEndPointIdList
EVENT	A0	17	RestartNwk_confirm
EVENT	A0	20	StopNwk_confirm
EVENT	A0	41	ZDP-NLME-NETWORK-DISCOVERY.Confirm
EVENT	A0	43	ZDP-NLME-PERMIT-JOINING.Confirm
EVENT	A0	47	ZDP-NLME-DIRECT-JOIN.Confirm
EVENT	A0	4B	ZDP-NLME-SYNC.Indication
EVENT	A0	4C	ZDP-NLME-SYNC.Confirm
EVENT	A0	80	ZDP-NWK_addr.response
EVENT	A0	81	ZDP-IEEE_addr.response
EVENT	A0	82	ZDP-NodeDescriptor.Response
EVENT	A0	83	ZDP-PowerDescriptor.Response
EVENT	A0	84	ZDP-SimpleDescriptor.Response
EVENT	A0	85	ZDP-Active_EP_rsp.response
EVENT	A0	86	ZDP-Match_Desc_resp.response
EVENT	A0	90	ZDP-Complex_Desc_rsp
EVENT	A0	91	ZDP-UserDescriptor.Response
EVENT	A0	92	ZDP-Discovery_Cache_response ZB1.1

TYPE	OpCode Group	OpCode	NAME
EVENT	A0	94	ZDP-User_Desc_conf
EVENT	A0	95	ZDP-System_Server_Discovery.response ZB1.1
EVENT	A0	96	ZDP-Discovery_store.response ZB1.1
EVENT	A0	97	ZDP-Node_Desc_store.response ZB1.1
EVENT	A0	98	ZDP-Power_Desc_store.response ZB1.1
EVENT	A0	99	ZDP-Active_EP_store.response ZB1.1
EVENT	A0	9A	ZDP-Simple_Desc_store.response ZB1.1
EVENT	A0	9B	ZDP-Remove_node_cache.response ZB1.1
EVENT	A0	9C	ZDP-Find_node_cache.response ZB1.1
EVENT	A0	A0	ZDP-ENDDEV_BIND.Resp
EVENT	A0	A0	ZDP-End_Device_Bind.response ZB1.1
EVENT	A0	A1	ZDP-BIND.Resp
EVENT	A0	A2	ZDP-UNBIND.Response
EVENT	A0	A3	ZDP-Bind_Register.response
EVENT	A0	A4	ZDP-Replace_Device.response ZB1.1
EVENT	A0	A5	ZDP-Store_Bkup_Bind_entry.response ZB1.1
EVENT	A0	A6	ZDP-Remove_Bkup_Bind_entry.response ZB1.1
EVENT	A0	A7	ZDP-Backup_Bind_Table.response ZB1.1
EVENT	A0	A8	ZDP-Recover_Bind_Table.response ZB1.1
EVENT	A0	A9	ZDP-Backup_Source_Bind.response ZB1.1
EVENT	A0	AA	ZDP-Recover_Source_Bind.response ZB1.1
EVENT	A0	B0	ZDP-Mgmt_Nwk_Disc.Response
EVENT	A0	B1	ZDP-Mgmt_Lqi.Response
EVENT	A0	B2	ZDP-Mgmt_Rtg.Response
EVENT	A0	B3	ZDP-Mgmt_Bind.Response
EVENT	A0	B4	ZDP-Mgmt_Leave.Response
EVENT	A0	B5	ZDP-Mgmt_Direct_Join.Response
EVENT	A0	B6	ZDP-Mgmt_Permit_Join.response ZB1.1
EVENT	A0	B7	ZDP-Mgmt_Cache.response ZB1.1
EVENT	A0	D0	ZDP-APSME-ESTABLISH-KEY.response
EVENT	A0	D3	ReadExtAddr.Confirm
EVENT	A0	D5	ZDP-APSME-ESTABLISH-KEY.Confirm
EVENT	A0	D6	ZDP-APSME-ESTABLISH-KEY.Indication
EVENT	A0	D7	ZDP-APSME-UPDATE-DEVICE.Indication
EVENT	A0	D8	ZDP-APSME-REMOVE-DEVICE.Indication
EVENT	A0	D9	ZDP-APSME-REQUEST-KEY.Indication
EVENT	A0	DD	ConfigurationAttrib.Confirm
EVENT	A0	DD	ConfigurationAttrib.Confirm
EVENT	A0	E1	AddressMapIndex.Confirm
EVENT	A0	E3	AddressMap.Confirm
EVENT	A0	E5	BeeStackSave.Confirmation
EVENT	A0	E6	EventOccurred.Indication
EVENT	A0	F1	ZDP-APSME-SWITCH-KEY.Indication
EVENT	A0	F2	ZDP-APSME-TRANSPORT-KEY.Indication
EVENT	A0	FB	SetApsSecMats.Confirm
EVENT	A0	FD	SetNwkSecMats.Confirm

TYPE	OpCode Group	OpCode	NAME
EVENT	A1	00	ReturnActiveNumOfEndPoints
EVENT	A1	01	FindEndPointDesc.Confirmation
EVENT	A1	02	EndPointIdsList
EVENT	A2	00	ZDP-NWK_addr_req.Request
EVENT	A2	01	ZDP-IEEE_addr_req.Request
EVENT	A2	02	ZDP-NodeDescriptor.Request
EVENT	A2	03	ZDP-PowerDescriptor.Request
EVENT	A2	04	ZDP-SimpleDescriptor.Request
EVENT	A2	05	ZDP-Active_EP_req.Request
EVENT	A2	06	ZDP-Match_Desc_req.Request
EVENT	A2	10	ZDP-Complex_Desc_req
EVENT	A2	11	ZDP-User_Desc_req.Request
EVENT	A2	12	ZDP-Discovery_Cache.request ZB1.1
EVENT	A2	13	ZDP-EndDeviceAnnounce.Request
EVENT	A2	14	ZDP-User_Desc_set
EVENT	A2	15	ZDP-System_Server_Discovery.Request
EVENT	A2	16	ZDP-Discovery_Store.Request
EVENT	A2	17	ZDP-Node_Desc_Store.Request
EVENT	A2	18	ZDP-Power_Desc_Store.Request
EVENT	A2	19	ZDP-Active_EP_Store.Request
EVENT	A2	1A	ZDP-Simple_Desc_Store.Request
EVENT	A2	1B	ZDP-Remove_Node_Cache.Request
EVENT	A2	1C	ZDP-Find_Node_Cache.Request
EVENT	A2	20	ZDP-End_Device_Bind.Request
EVENT	A2	21	ZDP-BIND.Request
EVENT	A2	22	ZDP-UNBIND.Request
EVENT	A2	23	ZDP-Bind_Register.Request
EVENT	A2	24	ZDP-Replace_Device.Request
EVENT	A2	25	ZDP-Store_Bkup_Bind_entry.Request
EVENT	A2	26	ZDP-Remove_Bkup_Bind_Entry.Request
EVENT	A2	27	ZDP-Backup_Bind_Table.Request
EVENT	A2	28	ZDP-Recover_Bind_Table.Request
EVENT	A2	29	ZDP-Backup_Source_Bind.Request
EVENT	A2	2A	ZDP-Recover_Source_Bind.Request
EVENT	A2	30	ZDP-Mgmt_Nwk_Disc.Request
EVENT	A2	31	ZDP-Mgmt_Lqi.Request
EVENT	A2	32	ZDP-Mgmt_Rtg.Request
EVENT	A2	33	ZDP-Mgmt_Bind.Request
EVENT	A2	34	ZDP-Mgmt_Leave.Request
EVENT	A2	35	ZDP-Mgmt_Direct_Join.Request
EVENT	A2	36	ZDP-Mgmt_Permit_Joining.Request
EVENT	A2	36	ZDP-NLME-PERMIT-JOINING.Request
EVENT	A2	37	ZDP-Mgmt_Cache.Request
EVENT	A2	39	ZDP-NLME-DIRECT-JOIN.Request
EVENT	A2	3C	ZDP-NLME-SYNC.Request
EVENT	A2	50	ZDP-NLME-NETWORK-DISCOVERY.Request

TYPE	OpCode Group	OpCode	NAME
EVENT	A2	CF	ZDP-APSME-ESTABLISH-KEY.Request
EVENT	A2	D1	ZDP-APSME-TRANSPORT-KEY.Request
EVENT	A4	D2	ReadExtAddr.Request
EVENT	A2	D3	ZDP-APSME-REQUEST-KEY.Request
EVENT	A2	D4	ZDP-APSME-SWITCH-KEY.Request
EVENT	A2	DA	ConfigurationAttributes.Init
EVENT	A2	DB	WriteExtAddr.Request
EVENT	A2	DC	StopNwk.Request
EVENT	A2	E0	RestartNwk.Request
EVENT	A2	E4	BeeStackSave.Request
EVENT	A2	F3	ZDP-APSME-UPDATE-DEVICE.Request
EVENT	A2	F4	ZDP-APSME-REMOVE-DEVICE.Request
EVENT	A2	FA	SetApsSecurityMaterials.Request
EVENT	A2	FC	SetNwkSecurityMaterials.Request
EVENT	A2	FE	Check-NeighborTable-ForRouting.Request
EVENT	A3	DF	StartNwk.Request
EVENT	A4	00	ZTCModeSelect.Confirmation
EVENT	A4	02	ZTCGetMode.Confirmation
EVENT	A4	04	ZTCEvent
EVENT	A4	05	ZTC_AFRreset.confirm
EVENT	A4	06	ZTC_APSReset.confirm
EVENT	A4	07	ZTC_SetAPSReady.confirm
EVENT	A4	DB	WriteExtAddr.Confirm
EVENT	A4	DF	StartNwk.confirm